

全国工程硕士专业学位教育指导委员会推荐教材

计算机网络安全

——协议、技术与应用

黄　河 编著　李伟琴 审核



http://www.tup.com.cn

清华大学出版社

全国工程硕士专业学位教育指导委员会推荐教材

计算机网络安全
——协议、技术与应用

黄　河　编著
李伟琴　审核

清华大学出版社
北　京

内 容 简 介

本书以 TCP/IP 网络安全协议为核心，全面、系统地论述计算机网络安全的协议、技术与应用等问题。全书共分为三个部分：网络安全基础部分，包括网络安全概述、密码学基础、数字认证技术和公钥基础设施等；网络安全协议部分，分层描述计算机网络各层的安全协议及其应用，包括网络层的 IPSec，传输层的 SSL/TLS，应用层的 S/MIME、PGP、SSH、DNSSEC、TSIG、SNMPv3 等；网络安全技术与应用部分，详细讲解防火墙、VPN、访问控制、入侵检测、系统审计等较为成熟的网络安全技术，同时还介绍了移动 IP 安全、无线网络安全、Web Service 安全等网络安全新技术。

本书可作为通信、计算机等相关专业的大学本科生和研究生教材，也可作为从事计算机网络与信息安全工作的工程技术人员和广大爱好者的参考书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989　13701121933

图书在版编目(CIP)数据

计算机网络安全——协议、技术与应用/黄河编著. —北京：清华大学出版社，2008.9
(全国工程硕士专业学位教育指导委员会推荐教材)
ISBN 978-7-302-18057-9

Ⅰ. 计…　Ⅱ. 黄…　Ⅲ. 计算机网络－安全技术－研究生－教材　Ⅳ. TP393.08

中国版本图书馆 CIP 数据核字(2008)第 098217 号

责任编辑：丁　岭　赵晓宁
责任校对：梁　毅
责任印制：

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185×230　　**印　张**：25.5　　**字　　数**：551 千字
版　　次：2008 年 9 月第 1 版　　**印　　次**：2008 年 9 月第1 次印刷
印　　数：1～　000
定　　价：　.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。
联系电话：010-62770177 转 3103　　产品编号：

序

软件质量具有功能性、可靠性、易用性、效率性、维护性和可移植性6个特性，可从软件的内部质量、外部质量和使用质量三个视角去考量。软件质量保证就是要求把质量嵌入到软件开发的生命周期全过程中，以保证软件的“生产”质量；软件测试是软件质量保证的一个关键手段，又是软件产品发布前的最终检验；对软件产品质量的评价是以量化的方式来说明软件质量的程度的。因此，软件质量保证、测试与评价三方面的内容是一个相互关联的体系。鉴于此，在上海实施了“软件质量专业技术职业资格”的培训与资格考试专家组的工作基础上，由于杨根兴教授长期从事软件质量保证、测试与评价研究和实践工作，因此，以他为主编写此书确可担当。该书主要特点如下。

1. 创新与继承相结合

软件质量随着软件工程学科的不断发展而推陈出新，该书内容既论述了行之有效的质量保证方法和技术，也在实践经验基础上总结出了一些重要内容，如风险管理、软件缺陷管理、测试用例的复用和面向应用的测试等。

2. 规范与整体相结合

通过对GB/T 16260、8566、17544、CMU/SEI CMMI等国家和国际标准的学习和研究，运用了这些标准中相关概念和过程的规范描述。既具有标准的依从性，又有从软件质量保证和软件测试两个方面较为深入和详细的阐述，形成了一个较为完整的体系。

3. 技术与管理相结合

软件质量保证的实践活动大多需要在软件企业中进行，虽然技术十分重要，而管理也非常重要。该书内容既论述技术和方法，也阐述了软件测试管理的内容和方法。在软件质量保证中，管理同样会出效益，也会出质量。

4. 理论与实践相结合

任何理论的存在，必有其实践背景。软件质量从重要性来讲，实践经验是第一位的。该

Preface

书从不同的侧面反映了我国在软件质量方面的研究成果和实践经验，使之理论和实践均能兼顾和融合。

以我毕生研究软件质量的经验，软件质量的保证、测试与评价是一大难题，特别是要提出一套符合中国文化理念的方法有待时日，尚需不断努力。因此，我们必须培养更多的软件质量保证和软件测试人才，共同努力，为中国软件产业的发展作出积极的贡献。

该书的出版，将会有益于读者掌握一门重要的技艺，有益于推动软件质量保证与测试的研究、教学、实践的进一步发展，有益于助推我国软件产业的发展。

朱三元

2007 年 9 月于上海

前 言

本书为全国工程硕士研究生教育核心教材，同时得到“北京市精品教材”项目的资助。本书由北京航空航天大学的黄河博士编著，李伟琴教授审核。

网络安全既有丰富的理论基础，又是实践性较强的一门学科。本书主要讲述网络环境下的信息安全技术，对于传统信息安全中的密码算法和密码协议等理论内容只做了简要介绍，着重描述了计算机网络安全的理论和实践知识，对于密码学的应用则贯穿在网络安全协议和技术中进行描述。为了使内容的组织具有系统性并便于读者理解，将教材内容划分为网络安全协议、技术和应用等不同层次进行论述。其中，网络安全协议部分是教材的核心内容，力争系统、全面地讲解 TCP/IP 网络安全协议，协议部分尽量结合其应用背景、实例和方法等加以论述；网络安全技术重点讲述防火墙、虚拟专用网、访问控制、入侵检测和系统审计等网络安全防御技术，同时也介绍了网络安全方面新的研究方向和技术。

教材中每章的实验部分列出了编者认为与教材内容配套的实验名称，其具体实验内容和方法读者可以根据需要自己选取。

教材的编写过程中得到诸多老师、同事及同学的帮助。北京航空航天大学的李伟琴教授对全书进行了审核并提出了许多宝贵的修改意见。北京航空航天大学软件学院的研究生周由胜、张凡、马心意和王隽竹，北京交通大学的研究生顾成杰，中国电力国际发展有限公司的陆路，路透中国科技有限公司的郑久丹，中国移动研究院的张鑫等同志也参与了本书的编写，在此表示诚挚的谢意。

由于作者水平所限，书中难免存在一些疏漏和错误，敬请广大读者批评指正。

编　者

2008 年 5 月



目录

第一部分　网络安全基础

Contents

第 4 章 公钥基础设施 /80

第二部分 TCP/IP 网络安全协议

第 5 章 网络层安全协议 /125

第三部分　网络安全技术与应用

第一部分

网络安全基础

第1章

网络安全概述

1.1 网络安全的概念及目标

网络安全是指对网络系统的硬件、软件及其中的数据实施保护，使网络信息不因偶然或恶意攻击而遭到破坏、更改或泄露，并且保证网络系统连续、可靠、正常地运行，保证网络服务不中断。

网络安全是一门涉及计算机科学、网络技术、通信技术、密码技术、信息安全技术、应用数学、数论和信息论等多种学科的综合性学科。

如图1.1所示，网络信息安全包括以下三个目标。

- 机密性(confidentiality)：指计算机系统的资源应该仅能由授权实体读取。
- 完整性(integrity)：指资源只能由授权实体修改。
- 可用性(availability)：指一旦用户得到访问某一资源的权限，该资源就应该能够随时为其使用。

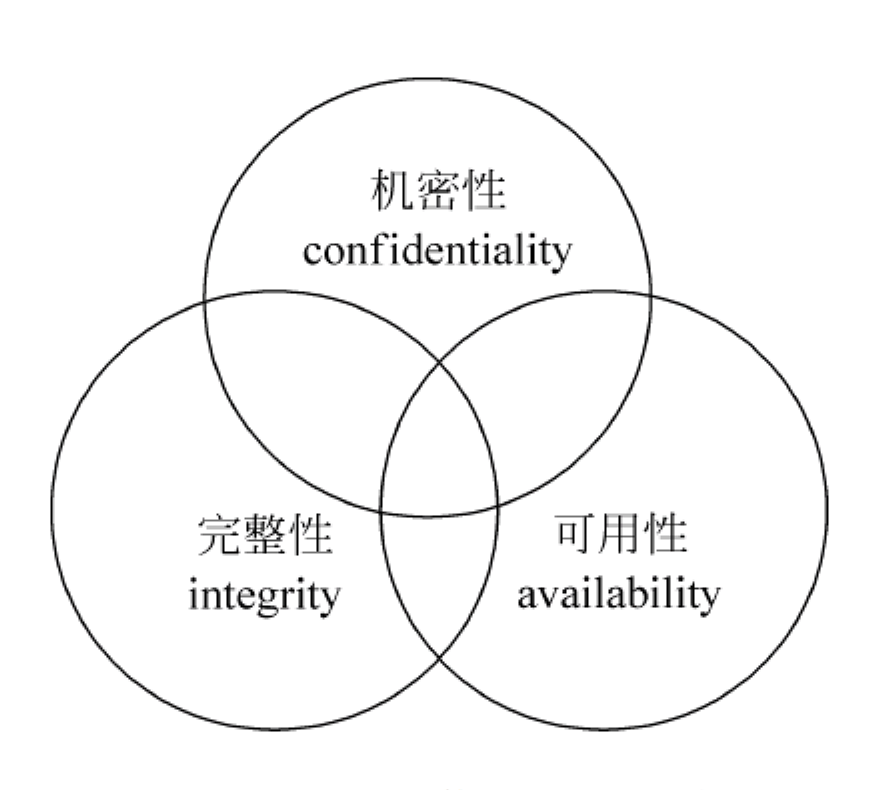


图1.1 网络信息安全目标

国际电信联盟(international telecommunication union，ITU)将网络安全定义为攻击、安全机制和安全服务三个部分。

其中，攻击(attack)是指损害机构所拥有信息的安全的行为；安全机制(security mechanism)是指设计用于检测、预防安全攻击或者恢复系统的方法；安全服务(security service)是指采用一种或多种安全机制以抵御安全攻击、提高机构的数据处理系统安全和信息传输安全的能力。安全机制和安全服务将在1.3节详述，下面简单分析针对网络和信息的攻击。

从攻击的动机和危害性上看，可以将网络攻击分为被动攻击和主动攻击两类。其中，被

动攻击是指进行网络窃听，截取数据包并进行分析，从中窃取敏感信息。被动攻击不会导致对系统中所含信息的任何改动，而且系统的操作和状态也不被改变，因此被动攻击主要威胁信息的机密性，被动攻击不易被检测。主动攻击是指意在篡改系统中所含信息或者改变系统的状态及操作，例如冒充、篡改、抵赖、非授权访问、非法登录、信息或网络服务破坏等，主动攻击可以被网络系统检测到。

从信息流的角度上看，网络中的数据受到四个方面的威胁，包括中断威胁、侦听威胁、修改威胁和伪造威胁。设信息是从源地址流向目的地址，那么正常的信息流向如图 1.2 所示。

(1) 中断威胁：如图 1.3 所示，中断威胁使得信息在传输过程中被阻断，无法正确到达目的地，导致正在使用的信息系统毁坏或不能正常使用，破坏系统的可用性。中断威胁的攻击手段包括切断网络通信线路、损坏网络服务和使文件系统瘫痪等。

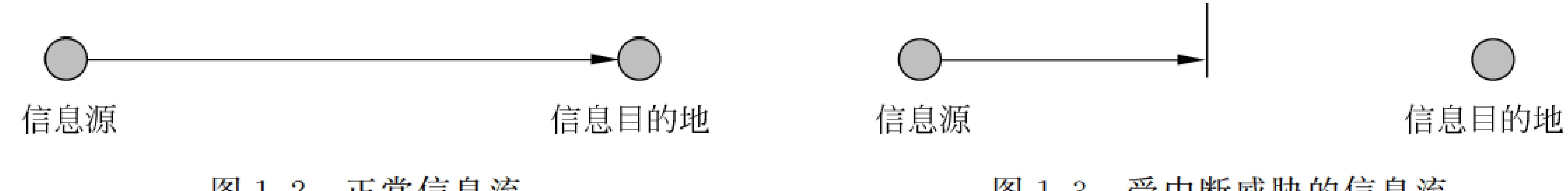


图 1.2　正常信息流　　　　图 1.3　受中断威胁的信息流

(2) 侦听威胁：如图 1.4 所示，在侦听威胁中，一个非授权方进入系统并获取资源，破坏系统的机密性。非授权方可以是一个人、一个程序或一台主机。侦听威胁的攻击手段包括搭线窃听，文件或程序的不正当复制等。

(3) 修改威胁：如图 1.5 所示，在修改威胁中，一个非授权方不仅介入系统而且对系统中的信息进行了修改，破坏系统的完整性。修改威胁包括改变数据文件、改变程序使之不能正确执行、修改报文内容等。

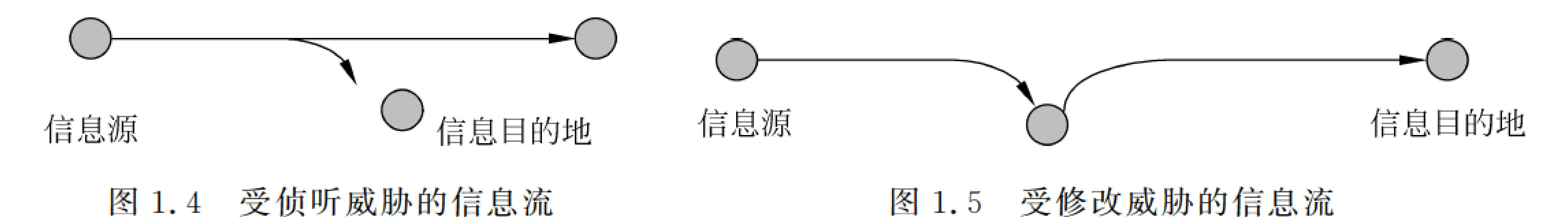


图 1.4　受侦听威胁的信息流　　　　图 1.5　受修改威胁的信息流

(4) 伪造威胁：如图 1.6 所示，在伪造威胁中，一个非授权方将伪造的客体插入到系统中，破坏信息的真实性。伪造威胁包括在网络中插入虚假信息，或者在文件中追加记录等。

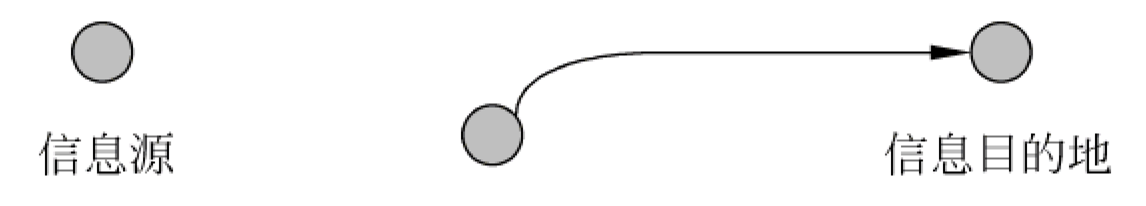


图 1.6　受伪造威胁的信息流

对应以上安全威胁，针对网络系统的攻击有多种，例如：

- 冒充攻击。指一个实体伪装成另一个实体，使网络中的信息遭受修改威胁和伪造威胁。例如，攻击者可以通过密码嗅探和猜测等方式获取合法用户的密码，然后冒充

该用户并利用该用户的权限对系统实施攻击。

- 重放攻击。指获取网络系统中的有效数据段，并以重播的方式对网络实施攻击，使网络中的信息遭受伪造攻击。例如，攻击者可以使用某种工具截获网络中的数据包，然后将该数据包重新发往目标地址。
- 修改攻击。指攻击者对网络中的数据包进行修改、延时和重排等，使网络中的信息遭受修改威胁。
- 拒绝服务攻击。指破坏网络设备或服务的正常运行，导致其无法提供正常服务，使网络中的信息遭受中断威胁。例如，攻击者可以发送大量垃圾信息使网络过载，导致网络和应用系统性能严重下降，破坏网络系统及其服务的可用性。

1.2 网络安全现状

计算机网络，特别是 Internet 的兴起和应用给人们的生活和工作带来了重大变化，随着计算机网络的普及，网络的应用向深度和广度不断发展。在网络给人们带来便利的同时，也带来了一些不容忽视的问题，网络信息的安全保密问题就是其中之一。随着时间的发展，各种网络安全问题也日益突出，图 1.7 给出了不同时间的已知威胁类型和数量。

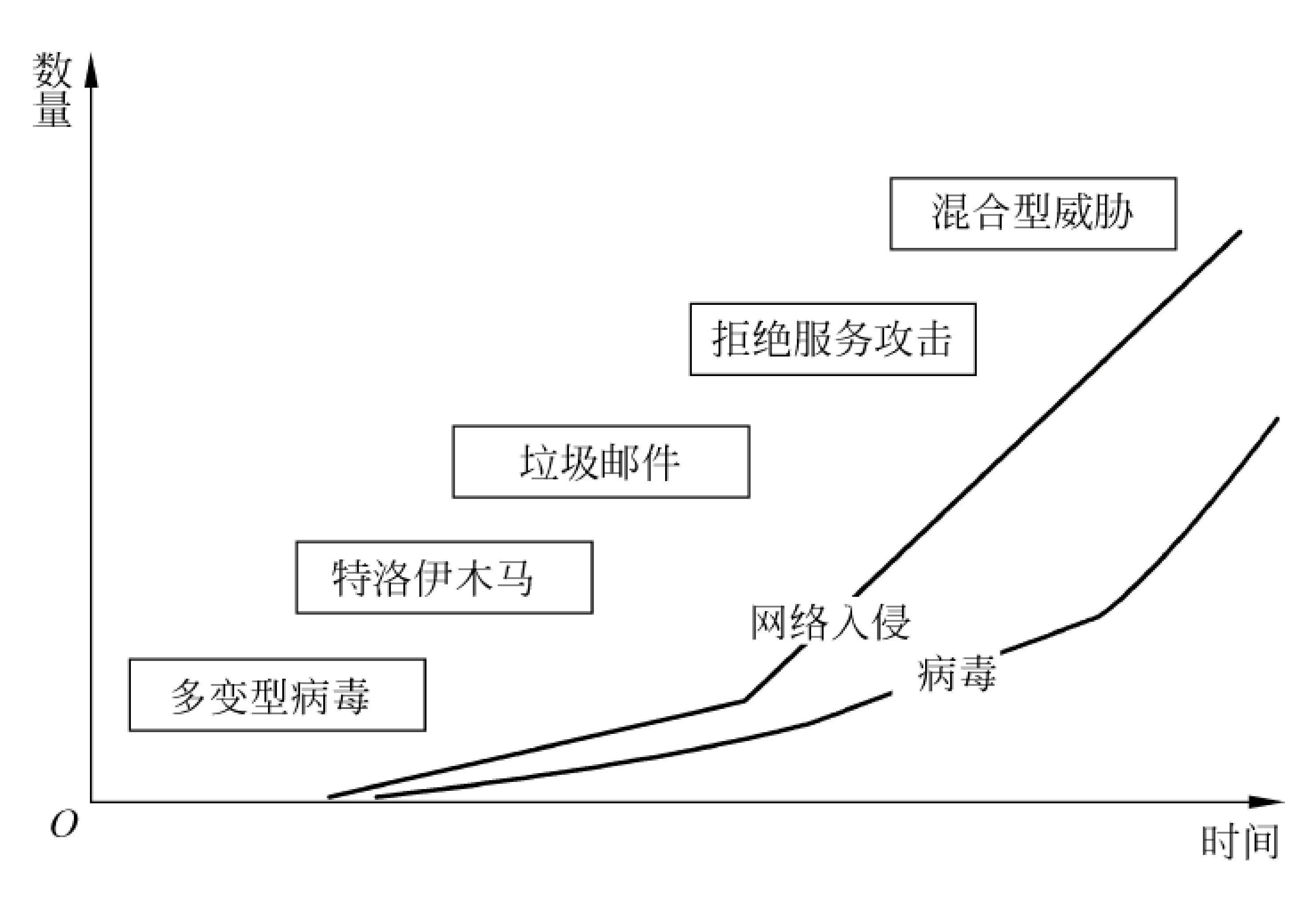


图 1.7 威胁种类及数量

计算机网络面临的威胁大体可分为两类：一是对网络中信息的威胁，二是对网络中设备的威胁。威胁计算机网络及其应用系统安全性的因素很多，有些因素可能是有意的，也可能是无意的；可能是人为的，也可能是自然的。归结起来，目前针对网络安全的威胁主要有如下三种。

① 人为的无意失误。例如因操作员安全配置不当造成的安全漏洞，用户安全意识不

强,用户密码选择不慎,用户将自己的账号随意转借他人或与别人共享等都会对网络安全带来威胁。

② 人为的恶意攻击。这是计算机网络所面临的最大威胁,敌手的攻击和计算机犯罪就属于这一类。此类攻击又可以分为主动攻击和被动攻击两类,这两种攻击均可对计算机网络造成极大的危害,并导致机密数据的泄露。

③ 网络软件的漏洞和"后门"(back door)。网络软件不可能是百分之百的无缺陷和无漏洞的,然而,这些漏洞和缺陷恰恰是黑客进行攻击的首选目标,曾经出现过的黑客攻入网络内部的事件大部分是因为安全措施不完善所招致的。另外,软件的"后门"一般是网络及其应用系统的设计人员为了某种开发或管理的需要而设置的,一般不为外人所知,但一旦"后门"被攻击者发现,其后果将不堪设想。另一类"后门"是攻击者在进入系统后刻意留下的,以便为日后的攻击或盗取资料提供便利。

Internet 的普及和应用使得这种基于网络漏洞的攻击变得非常频繁,造成的损失巨大。Internet 起源于 1969 年的 ARPANet,最初用于军事目的,1993 年开始用于商业应用,进入快速发展阶段。到目前为止,Internet 已经覆盖了世界上大多数国家和地区的数千万台计算机,用户数量庞大。这个全球范围的 IP 网络,连接着不同的网络资源、政府资源、教育资源以及商贸资源。Internet 上各种应用及其的复杂程度日益增加。

Internet 的开放性、异构性、多数应用是客户机/服务器模式、网络允许部分匿名用户等的设计方式和特点,使得 Internet 上的漏洞很快可以被发现并迅速传播,针对各种网络漏洞的攻击手段和攻击工具同样也是开放的。普通用户可以得到各种攻击工具,对攻击行为感兴趣,或喜欢尝试和冒险,以及从攻击行为中牟利、以攻击为职业的人随时可以利用这些开放的工具对网络进行攻击,窃取机密信息、对网络资源进行破坏和更改等。Internet 漏洞的客观存在以及其特性和设计原则使得基于 Internet 的攻击难以防范。

1988 年 11 月 3 日,第一个"蠕虫"病毒被放到 Internet 上,数小时之内,数千台机器被传染,Internet 几乎陷入瘫痪。"蠕虫"的作者 Robert Morris J. r 被判有罪,接受三年监护并被罚款。"Morris 蠕虫"的出现改变了许多人对 Internet 安全性的看法。一个单纯的程序有效地摧毁了数千台机器,那一天同时标志着 Internet 安全性研究的开始。

图 1.8 给出了一个利用 Web 服务进行攻击的例子。Internet 上的安全漏洞包括操作系统的、网络的和应用的,这些漏洞都可能被攻击者利用进行攻击或作为攻击其他机器的手段。例如木马程序可以人为安放在恶意的代码中,然后嵌入在 Web 数据服务器的网页中,从而对网络上的主机实施攻击。

图 1.9 中是 TCP 会话劫持的示意图。一个客户程序通过 TCP 正在与一台服务器进行通信。攻击者可以使用 ARP(address resolution protocol)来截获和重定向客户与服务器之间的数据流,使之经过攻击者的机器。然后攻击者假冒客户机身份向服务器启动连接请求,使用的序列号为 Attack_SEQ。服务器向真正的客户端发送应答,应答序列号为 Attack_SEQ+1,自身序列为 New_SVR_SEQ。由于真实客户端接收到的应答序列号不符,应答包

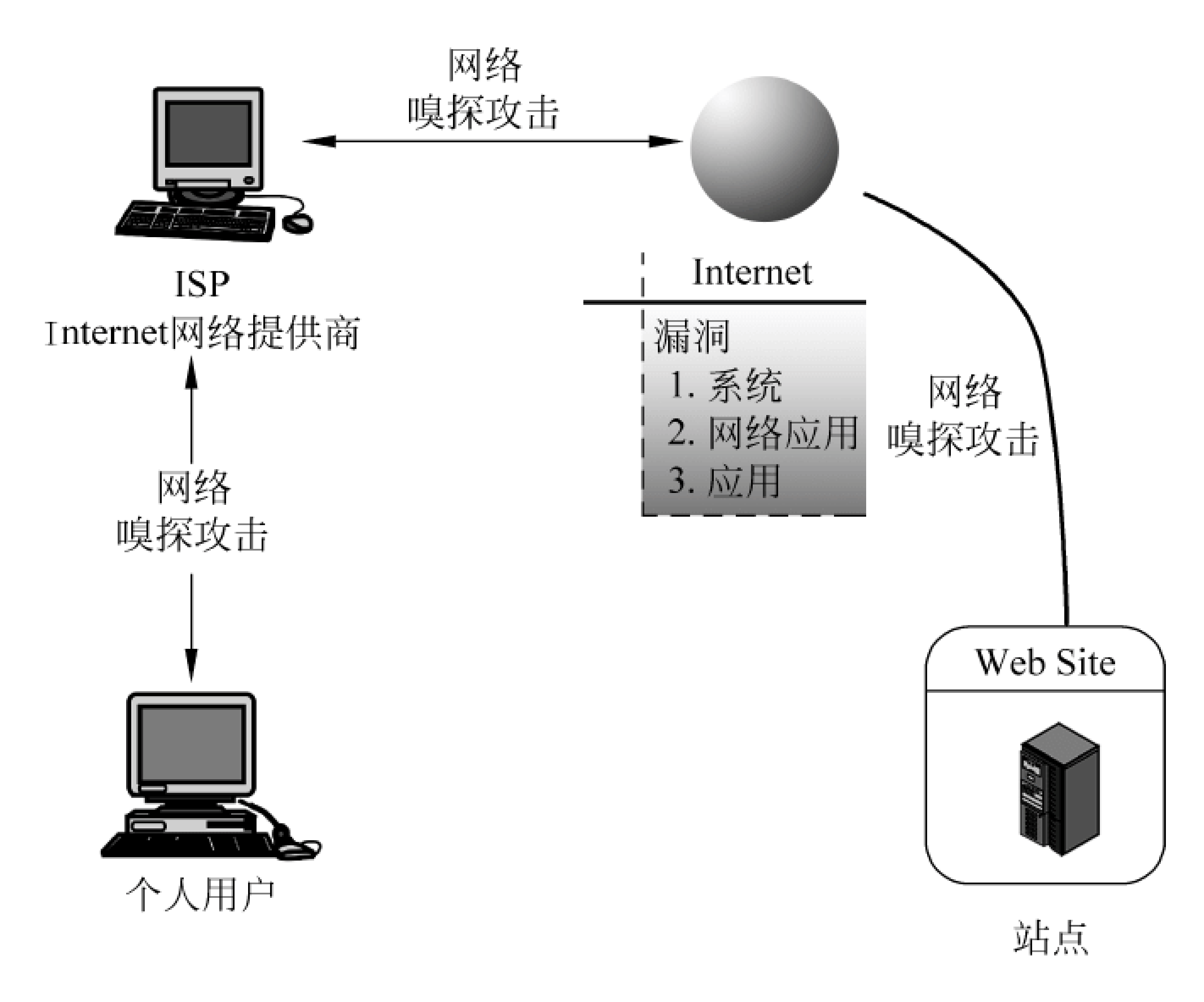


图 1.8　Internet 服务中存在的安全漏洞

被客户端忽略。攻击者再次假冒客户端进行应答,应答序列号为 New_SVR_SEQ+1。结果,服务器也处于连接建立状态。但是由于客户和服务器之间的序列号不一致,它们之间并不能真正地进行通信,各自发往对方的数据包都将被忽略,TCP 会话被攻击者劫持。

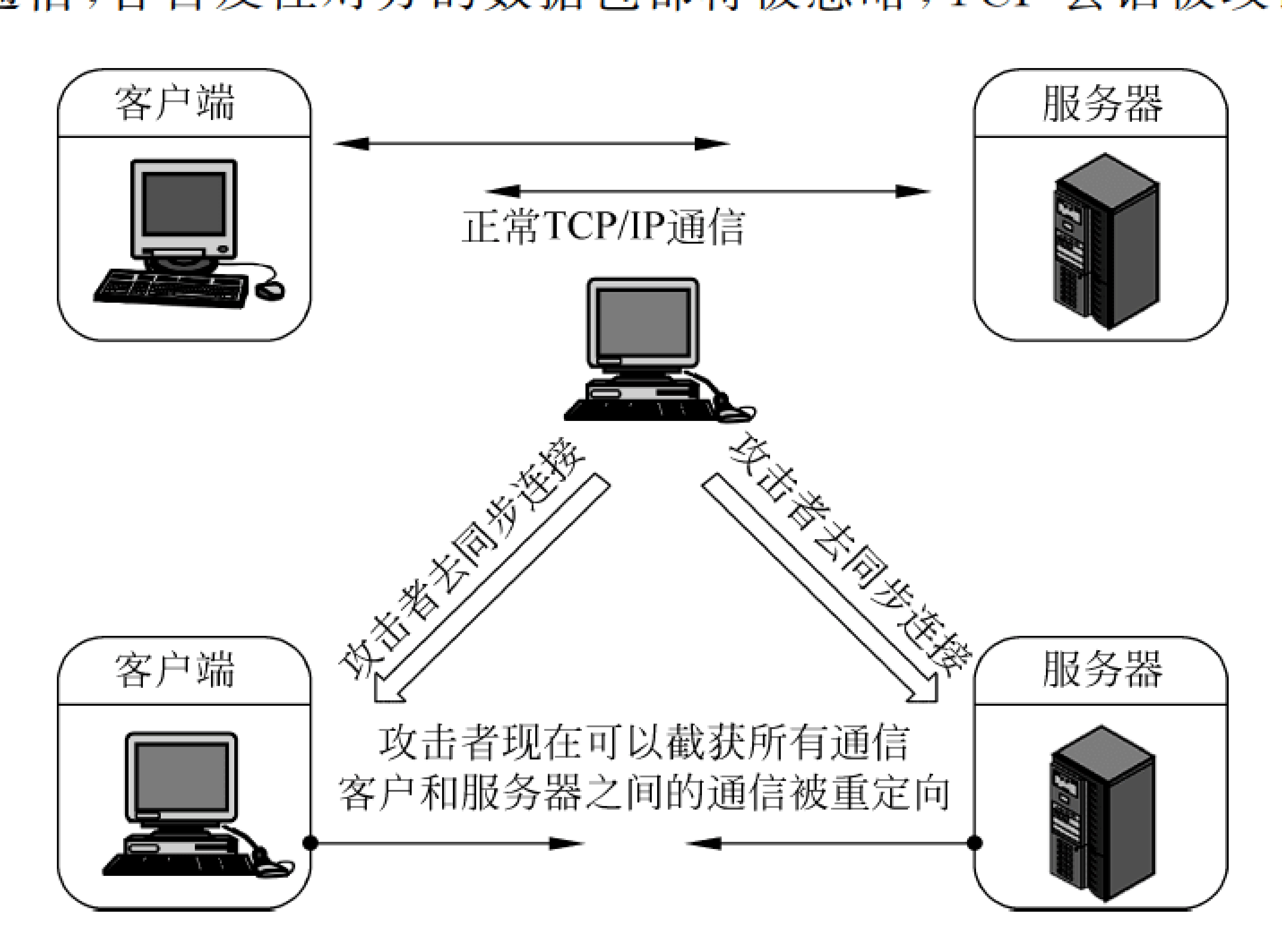


图 1.9　TCP 会话劫持

内部网络同样面临安全威胁,内部人员对网络或系统的威胁可归为两类,一是利用网络实施危害系统安全性的各种行为,如未授权访问、密码嗅探和盗取、IP 地址欺骗等;二是对网络中的各种资源,特别是涉密的文档进行非法打印、复制、转移等。由于内部人员一般是拥有一定权限的合法用户,他们本身具备访问网络的权限,因此这种基于内部网络的攻击更

加难以防范,需要进行严格的访问控制、内部资源的分级和控制等来提高内部网络的安全性。

1.3 ISO/OSI 网络安全体系

ISO/OSI 参考模型主要从安全策略、安全服务、安全机制和安全管理等方面描述网络安全体系结构。

1.3.1 安全策略

安全策略指在一定的环境内,为保证网络提供一定级别的安全保护能力所提出的、系统应该遵守的一系列规则。可以将网络安全策略分为物理安全策略、访问控制策略、信息加密策略以及网络安全管理策略等。

1. 物理安全策略

物理安全策略的目的是保护计算机系统、网络服务器、打印机等硬件实体和通信链路免受自然灾害、人为破坏和搭线攻击;验证用户的身份和使用权限、防止用户越权操作;确保计算机系统有一个良好的电磁兼容工作环境;建立完备的安全管理制度,防止非法进入计算机控制室和各种偷窃、破坏活动的发生。

抑制和防止电磁泄漏是物理安全策略的一个重要问题。目前主要防护措施有两类:一类是对传导发射的防护,主要采取对电源线和信号线加装性能良好的滤波器,减小传输阻抗和导线间的交叉耦合等方法。另一类是对辐射的防护,这类防护措施又可分为采用各种电磁屏蔽和干扰等。

2. 访问控制策略

访问控制是网络安全防范和保护的重要策略,它的主要任务是保证网络资源不被非法使用和非法访问。它也是维护网络系统安全、保护网络资源的重要手段,可以说是保证网络安全最重要的核心策略之一。下面举例说明几种访问控制策略。

- 入网访问控制。入网访问控制为网络访问提供第一层控制。它控制哪些用户能够登录到服务器并获取网络资源,控制允许用户入网的时间和允许他们在哪些工作站入网。用户的入网访问控制的实施可分为三个步骤:用户名的识别与验证、用户密码的识别与验证、用户账号的默认限制检查。
- 网络的权限控制。网络的权限控制是针对网络非法操作所提出的一种安全保护措施。网络管理员应当为用户指定适当的访问权限,这些访问权限控制着用户对服务

器的访问。用户和用户组被赋予一定的权限,规定他们可以访问哪些目录、子目录、文件和其他资源。可以指定用户对这些文件、目录和设备能够执行哪些操作。

- 目录级安全控制。网络应允许控制用户对目录、文件和设备的访问。用户在目录一级指定的权限对所有文件和子目录有效,用户还可进一步指定对目录下的子目录和文件的权限。
- 属性安全控制。当使用文件、目录和网络设备时,网络系统管理员应给文件、目录等指定访问属性。属性安全控制可以将给定的属性与网络服务器的文件、目录和网络设备联系起来。属性安全在权限安全的基础上提供更进一步的安全性。网络上的资源都应预先标出一组安全属性。用户对网络资源的访问权限对应一张访问控制表,用以表明用户对网络资源的访问能力。
- 网络服务器安全控制。网络允许在服务器控制台上执行一系列操作。用户使用控制台可以装载和卸载模块,可以安装和删除软件等。网络服务器的安全控制包括设置密码锁定服务器控制台,以防止非法用户修改、删除重要信息或破坏数据;还可以设定服务器登录时间限制、非法访问者检测和关闭的时间间隔等。
- 网络监测和锁定控制。网络管理员应对网络实施监控,服务器应记录用户对网络资源的访问,对非法的网络访问,服务器应以图形或文字或声音等形式报警,以引起网络管理员的注意。如果未授权人员试图进入网络,网络服务器应自动记录企图其尝试进入网络的次数,如果非法访问的次数达到设定数值,那么该账户将被自动锁定。
- 网络端口和节点的安全控制。网络中服务器的端口往往使用自动回呼设备、静默调制解调器加以保护,并以加密的形式来识别节点的身份。自动回呼设备用于防止假冒合法用户,静默调制解调器用以防范黑客的自动拨号程序对计算机进行攻击。还可以对服务器端和用户端采取控制,例如规定用户必须携带证实身份的验证器(如智能卡、磁卡和安全密码发生器)。在对用户的身份进行验证之后,才允许用户进入系统。然后,用户端和服务器端再进行相互认证。
- 防火墙控制。防火墙在网络边界上通过建立起来的相应网络通信监控系统来隔离内部和外部网络,以阻止来自外部网络的入侵。目前的防火墙主要有以下三种类型,即包括过滤防火墙、代理防火墙和双穴主机防火墙。

3. 信息加密策略

信息加密的目的是保护计算机网络的数据、文件、密码和控制信息,保护网络上传输的数据。网络加密常用的方法有链路加密、端到端(端点)加密和节点加密等。链路加密的目的是保护网络节点之间的链路信息安全;端到端加密的目的是对源端用户到目的端用户的数据提供保护;节点加密的目的是对源节点到目的节点之间的传输链路提供保护。用户可根据网络系统的安全性需求选择上述加密方式。信息加密过程由加密算法来实施,可以将

加密算法分为常规密码算法(对称密码)和公钥密码算法(非对称密码)。

- 常规密码

收信方和发信方使用相同的密钥,即加密密钥和解密密钥是相同或等价的。其优点是有很强的保密强度,但其密钥必须通过安全的途径传送。因此,密钥管理成为影响系统安全的重要因素。

- 公钥密码

收信方和发信方使用不同的密钥进行加密和解密,而且几乎不可能从加密密钥推导出解密密钥。其优点是可以适应网络的开放性要求,且密钥管理问题也较为简单,尤其可方便地实现数字签名和身份认证。但其算法复杂,加解密数据的速率较低。尽管如此,随着现代电子技术和密码技术的发展,公钥密码算法和数字证书成为一种应用广泛的身份认证和加密技术。

在实际应用中,人们通常将常规密码和公钥密码结合在一起使用。

4. 网络安全管理策略

在网络安全中,除了采用上述技术措施之外,加强网络的安全管理,制定有关规章制度,对于确保网络安全、可靠地运行,将起到十分有效的作用。安全管理策略从管理角度保障网络安全性,包括:指定信息安全政策、进行安全风险评估、安全控制目标与方式选择、制定安全规范、进行职工安全意识培训等多种措施和方法。BS7799(ISO/IEC 17799),即《信息安全管理体系标准》和 ISMS(information security management system)针对安全管理进行了详细描述。

5. 安全策略的实施

安全策略是设计一个安全的网络和应用系统的基础,网络和应用系统的建设应该围绕安全策略的制定和实施同时进行,以满足系统的安全性需求。如图 1.10 所示,一个安全的网络和应用系统的实施过程可简单描述如下。

首先,应该对网络及其应用系统进行安全需求分析和现状分析。例如,可以对网络的资产、脆弱性及安全威胁进行分析,进行网络安全风险分析,得出网络面临的安全风险。在风险分析的基础上,结合对网络系统进行的实际安全性需求,制定网络安全策略,即进行安全策略的设计。在策略设计的基础上,结合各种技术手段和安全设施,进行安全策略的实施。最后,对安全策略的实施结果进行评价和反馈,根据新的安全需求和安全风险,重新制定和实施策略,并不断循环。

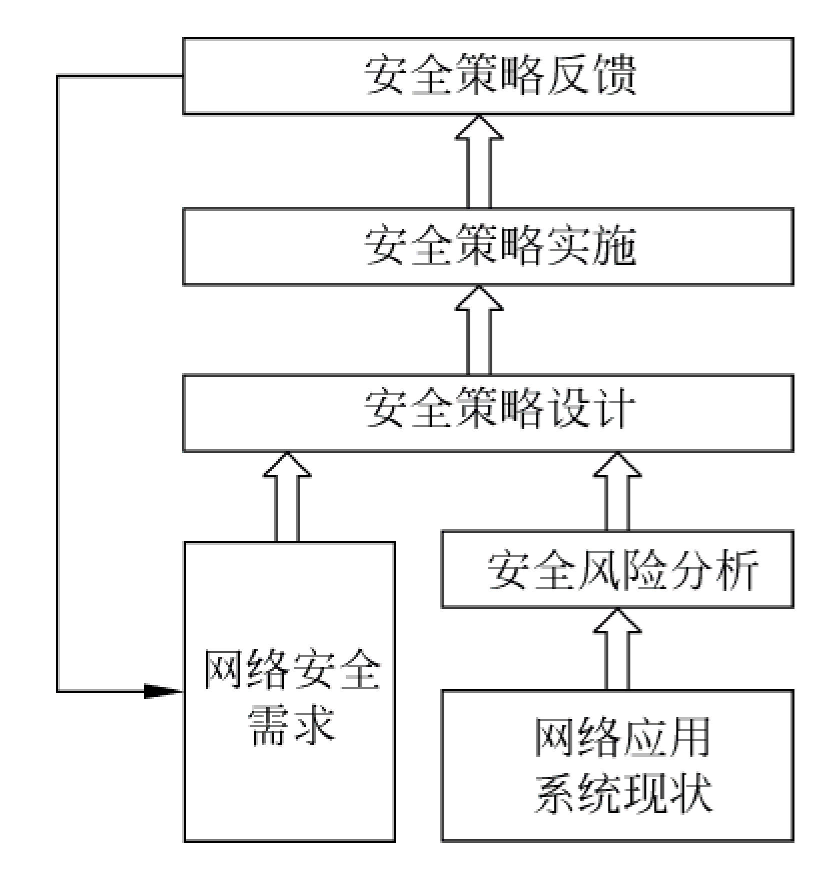


图 1.10 网络安全策略的设计和实施过程

1.3.2 安全服务

安全服务主要涉及如下内容。

- 身份认证(或鉴别,authentication)。确认用户、主机及数据源的身份。
- 访问控制(access control)。防止未经授权的数据存取行为,以身份认证为基础。
- 数据机密性(data confidentiality)。防止信息泄露至未授权实体。
- 数据完整性(data integrity)。保护数据以防止其内容遭篡改。
- 不可否认性(抗抵赖性,non-repudiation)。一旦事务结束,有关各方都不能否认自己参与过该事务。
- 可用性(availability)。系统具有向其合法用户提供资源或服务的能力。

以下针对上述内容具体说明。

1. 身份认证

身份认证包括对等实体认证和数据源认证。

如图 1.11 所示,对等实体认证由 n 层协议提供,$n+1$ 层实体可确信其对等实体是其所信赖之实体。对等实体认证应该保证某实体没有企图冒充别的实体,且没有非法重放以前的某个连接。

数据源认证在通信的某个环节,确认数据是由某个特定发送者发送的。它对数据的来源提供认证,但不提供防止数据单元被复制或篡改的安全服务。数据源认证包括两方认证(通信双方进行单向或双向认证)和基于第三方的认证(通信双方利用第三方进行身份鉴别)两种方式。

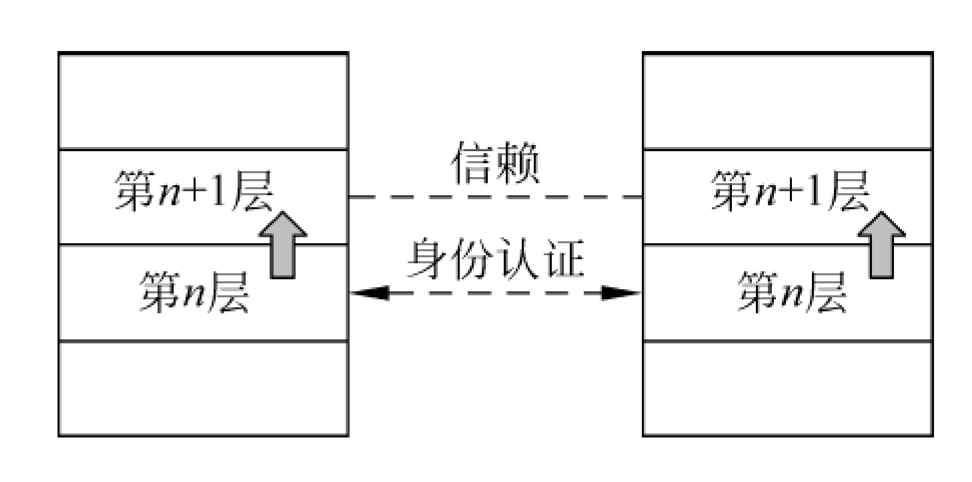


图 1.11 对等实体认证

2. 访问控制

访问控制采用管理或技术手段保证网络资源不被未授权使用,以实现访问控制策略。访问控制中主要包括三个实体:访问方,即主体(subject);被访问方,即客体(object),以及访问权限(access right)。系统根据主体对客体的访问权限决定是否允许执行该访问。

3. 数据机密性

数据机密性包括连接机密性、无连接机密性、选择字段保密性(向指定数据单元的某些字段提供机密性服务)以及业务流保密性(防止通过分析业务流得到机密信息)。其所涉及的加密技术有对称密钥算法、非对称密钥算法或公钥算法、哈希函数(hash function)和数字签名等。

4. 不可否认性

不可否认性涉及如下两个方面：

数据源的抗抵赖性，是指向数据接收者提供数据来源的证据，以防止发送者否认曾发送该数据及其内容。

传递过程的抗抵赖性是指向数据发送者提供数据已到达目的地的证据，以防止信息接收者否认曾接收该数据及其内容。

5. 数据完整性

数据完整性是数据本身的真实性证明。数据完整性服务对传输中的数据流进行验证，保证发送信息和接受信息的一致性。在使用密码技术进行验证时，一般使用非线性单向函数求出鉴别码，即 MAC(message authentication code)。使用单向散列函数(哈希函数)进行验证时，将输出的定长字符串作为对数据完整性的鉴别码。

1.3.3 安全机制

安全机制主要包括：加密机制、数字签名机制、访问控制机制、数据完整性机制、交换鉴别机制、业务流量填充机制、路由控制机制以及公证机制。以下分别说明。

1. 加密机制

加密是提供数据机密性服务最常用的方法。如前所述，按密钥类型划分，加密算法可分为对称密钥和非对称密钥加密算法两种；按密码体制分，可分为序列密码和分组密码算法两种。用加密的方法与其他技术相结合，可以提供数据的机密性和完整性服务。

2. 数字签名机制

数字签名机制可以解决通信双方发生争执时可能产生的如下安全问题。

- 否认：发送者事后不承认自己发送过某份文件。
- 伪造：接收者伪造一份文件，声称它发自发送者。
- 假冒：网络上的某个用户冒充另一个用户接收或发送信息。
- 篡改：接收者对收到的信息进行部分篡改。

3. 访问控制

访问控制是按事先确定的规则决定主体对客体的访问是否合法的安全机制。当一个主体试图非法访问一个未经授权的客体时，该机制将拒绝这一请求，并向审计跟踪系统报告该事件。

4. 数据完整性机制

数据完整性机制提供数据完整性服务，包括两种形式：一种是数据单元的完整性，另一种是数据单元序列的完整性。数据单元完整性包括两个过程，一个过程发生在发送实体，另一个过程发生在接收实体。

保证数据完整性的一般方法是发送实体在一个数据单元上加一个标记，这个标记就是MAC 鉴别码或哈希值，它本身可以是经过加密的。接收实体计算对应的标记，并将所产生的标记与接收的标记相比较，以确定在传输过程中数据没有被修改过。

数据单元序列的完整性要求数据编号的连续性和时间标记的正确性，以防止假冒、丢失、重发、插入或修改数据。

5. 交换鉴别机制

交换鉴别是以交换信息的方式来确认实体身份的机制。用于交换鉴别的基本技术有密码和密码技术两种。其中密码由发送方实体提供，接收方实体进行验证。密码鉴别技术是将交换的数据加密，只有合法用户才能解密，得出有意义的明文。

在许多情况下，这种密码和密码技术与许多其他技术一起使用，包括时间标记和同步时钟；双方或三方“握手”；数字签名和公证机构以及利用实体的特征或所有权进行鉴别，例如使用指纹和身份卡等。

6. 业务流量填充机制

这种机制主要是对抗非法入侵者在线路上监听数据并对其进行流量和流向分析。采用的方法一般由保密装置在无信息传输时，连续发出伪随机序列，使得侦听者不知哪些是有用信息、哪些是无用信息。

7. 路由控制机制

在一个大型网络中，从源节点到目的节点可能有多条路径，有些路径可能是安全的，而另一些路径是不安全的。路由控制机制可使信息发送者选择特殊的路由，以保证数据安全。

8. 公证机制

网络中的用户并不都是可信的，同时也可能由于系统故障等原因使信息丢失、延迟，这很可能引起责任问题，为了解决这个问题，就需要有一个各方都信任的实体——公证机构提供公证服务，仲裁出现的问题。

一旦引入公证机制，通信双方进行数据通信时必须经过这个机构来转换，以确保公证机构能得到必要的信息，供以后仲裁时使用。

1.3.4 安全管理

安全管理是指通过实施一系列安全政策，对系统和网络上的操作进行必要的控制和管理，包括系统安全管理、安全服务管理和安全机制管理等。

- 系统安全管理涉及 OSI 整体安全环境的管理，包括总体安全策略管理、OSI 安全环境之间的安全信息交换、安全服务管理、安全机制管理、安全事件管理、安全审计管理以及安全恢复管理等。
- 安全服务管理涉及特定安全服务的管理，包括对某种安全服务定义其安全目标、指定安全服务可使用的安全机制、管理及调用适当的安全机制。
- 安全机制管理涉及特定安全机制的管理，包括密钥管理、加密管理、数字签名管理、访问控制管理、数据完整性管理、认证(鉴别)管理、业务流量填充管理以及公证管理。

每种安全机制和安全服务可以抵御一种或几种攻击，图 1.12 说明了安全攻击、安全机制和安全服务之间的关系。

释放消息内容	流量分析	伪装	重放	更改消息	拒绝服务	安全攻击 / 安全机制 / 安全服务	加密	数字签名	访问控制	数据完整性	认证交换	流量填充	路由控制	公证
		√				对等实体认证	√	√			√			
		√				数据源认证	√	√						
		√				访问控制			√					
√						机密性	√						√	
	√					流量机密性	√					√	√	
			√	√		数据完整性	√	√		√				
						非否认服务		√		√				√
					√	可用性				√	√			

图 1.12 安全攻击、安全机制和安全服务之间的关系

例如，对等实体认证服务可以通过加密、数字签名和认证交换(交换鉴别)等安全机制来提供，该服务可以抵御伪装攻击。数据完整性服务可以使用加密、数字签名、数据完整性机制等来实现，该服务可以提供对消息的更改和重放攻击。

1.4　典型网络安全模型

1.4.1　动态自适应网络模型

为了有效实施网络安全防护，人们提出了 P2DR(policy protection detection response)模型。

如图 1.13 所示，模型以安全策略为核心，将安全问题归结为防护(protection)、检测(detection)和响应(response)不断循环的三个步骤。认为安全问题需要通过风险分析、执行策略、系统实施和漏洞检测与实时响应等一系列举措来解决。所有的行为均以安全策略作为依据，经过不断循环，逐步完善系统的安全性。其特点如下。

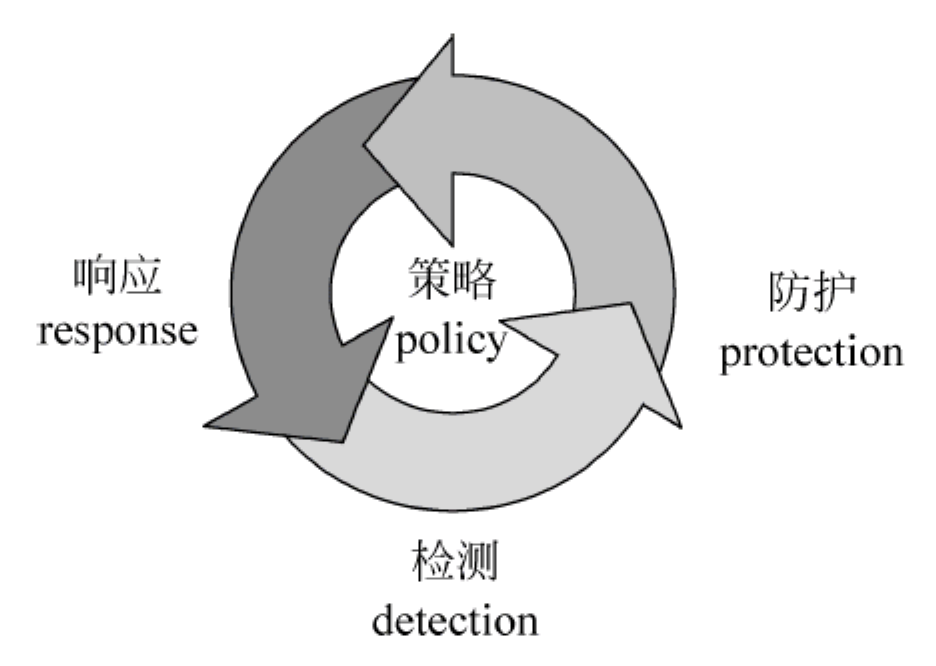


图 1.13　P2DR 模型

- 安全管理的持续性、安全策略的动态性：实时监视网络活动，发现威胁和弱点后即时修复系统漏洞。
- 可测性(可控性)：通过不断对网络系统进行评估而把握系统风险，及时弱化、堵塞安全漏洞。
- 时间性：确定防护时间、检测时间及响应时间之间的关系，保证入侵行为能够被即时检测并处理。

P2DR 模型使得安全防护(如采用防火墙、操作系统身份认证和加密等手段)、检测(如漏洞评估、入侵检测等)、响应三个部分在安全策略的统一控制和指导下形成一个动态的信息安全生命周期。

P2DR 模型突出了时间的概念，可以用下面的公式说明。

$$P_t > D_t + R_t$$

其中，P_t 代表被保护系统设置各种保护方式后的有效防护时间；D_t 代表从入侵者开始发动入侵到系统能检测到入侵所花费的时间；R_t 代表从发现入侵行为到系统能够做出响应，并将系统调整到新的安全状态的时间。针对要保护的目标，如果上述公式成立，表示防护时间大于检测时间加上响应时间，也就是说，在入侵者危害信息安全目标之前，其入侵行为可以被检测并及时处理，说明该防护系统是有效的。

P2DR 模型突破了传统安全模型的静态束缚，在 P2DR 模型的基础上，人们又根据自己的实际需求作了一定的修改和补充。比如有人认为恢复工作很重要，需要强调，于是有了 PDRR 模型(protection，detection，response and recovery)。

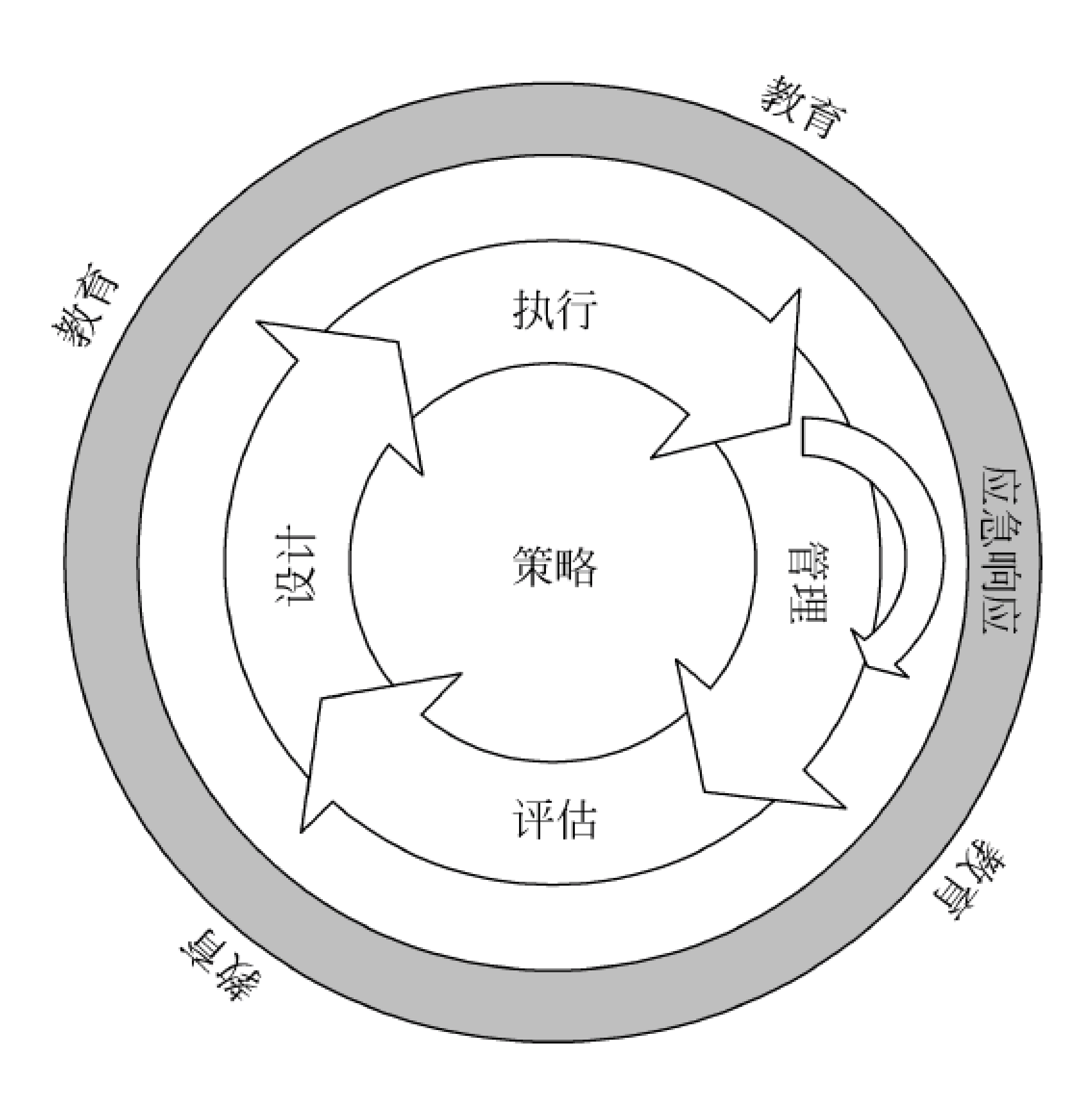


图 1.14　PADIMEE 模型

美国互联网安全系统公司 ISS 基于 P2DR，提出了自适应性网络安全模型(adaptive network security model，ANSM)，并联合其他厂商组成 ANSM 联盟，试图在此基础上建立全球网络安全标准。ISS 以安全服务方法论为基准，通过对技术和业务需求分析，以及对客户信息安全的“生命周期”考虑，提出 PADIMEE 模型。该模型在 7 个方面体现了信息系统安全的持续循环，分别是策略(policy)、评估(assessment)、设计(design)、执行(implementation)、管理(management)、应急响应(emergency response)和教育(education)，如图 1.14 所示。

1.4.2　APPDRR 模型

网络安全的动态特性在 P2DR 模型中得到了一定程度的体现，主要是通过入侵检测和响应完成网络安全的动态防护。但 P2DR 模型不能描述网络安全的动态螺旋上升过程。为了能够比较确切地描述网络安全的本质规律，人们对 P2DR 模型进行了修正和补充，在此基础上提出了 APPDRR 模型。

如图 1.15 所示，APPDRR 模型认为网络安全由风险评估(assessment)、安全策略(policy)、系统防护(protection)、动态检测(detection)、实时响应(reaction)和灾难恢复(restoration)6 个过程构成。

根据 APPDRR 模型，网络安全的第一个重要环节是风险评估，通过风险评估，掌握网络安全面临的风险信息，进而采取必要的处置措施，使信息组织的网络安全水平呈现动态螺旋上升的趋势。网络安全策略是 APPDRR 模型的第二个重要环节，它起着承上启下的作用：一方面，安全策略应当随着风险评估的结果和安全需求的变化做相应的更新；另一方

面，安全策略在整个网络安全工作中处于原则性的指导地位，其后的检测、响应等环节都应该在安全策略的基础上展开。系统防护是安全模型中的第三个环节，体现了网络安全的静态防护措施。接下来是动态检测、实时响应和灾难恢复三个环节，体现了安全动态防护和安全入侵、安全威胁的对抗性特征。

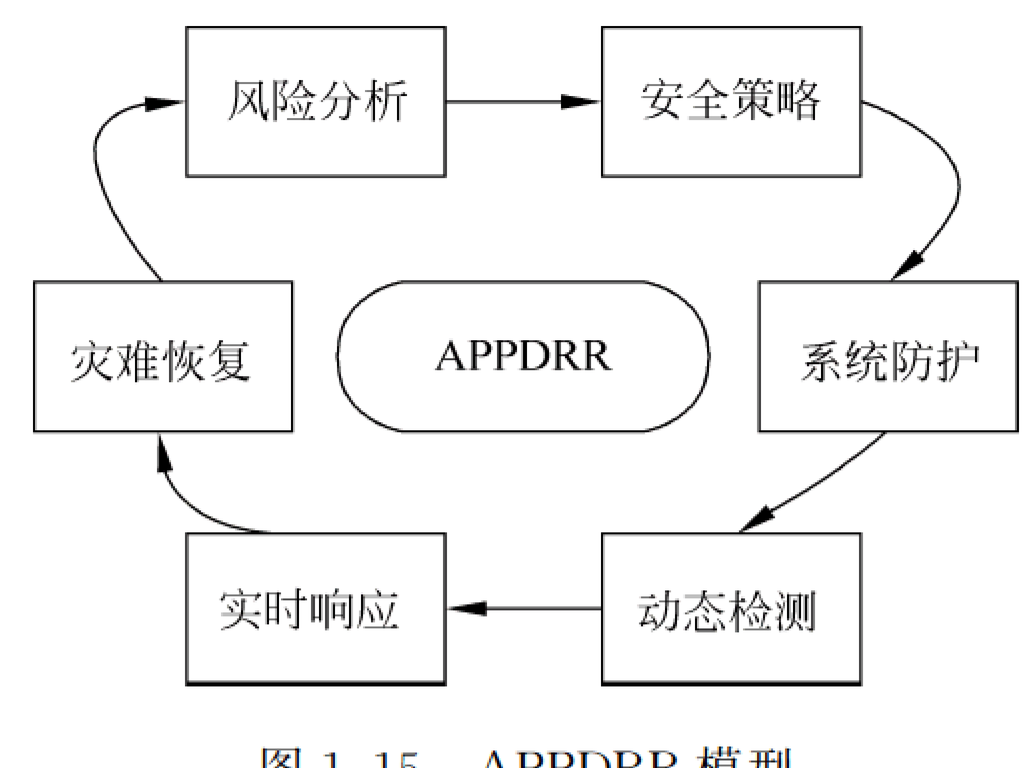


图 1.15　APPDRR 模型

APPDRR 模型还隐含了网络安全的相对性和动态螺旋上升的过程，即不存在百分之百的静态的网络安全，网络安全表现为一个不断改进的过程。通过风险评估、安全策略、系统防护、动态检测、实时响应和灾难恢复 6 个环节的循环流动，使网络的安全性逐渐得以完善和提高。

1.4.3　分层的网络安全解决方案

网络协议是分层的，因此对于网络安全问题，也可以从分层的角度出发，使用分层的安全协议解决网络安全问题。

根据 TCP/IP 参考模型，网络协议分为应用层、传输层、网络层和接口层。ISO/OSI 模型则将网络协议划分为应用层、表示层、会话层、传输层、网络层、链路层和物理层。结合 TCP/IP 和 OSI 参考模型，可以按照应用层、传输层、网络层、链路层和物理层进行网络安全防护体系的设计。

每层可以采用独立的安全协议和技术，也可以联合使用其他层的安全技术。例如，对于数据链路层安全，可以采用链路加密方式，确保机密性。网络物理安全的目的是保护计算机设备、网络设施以及传输介质和媒体免遭地震、水灾、火灾、有害气体和其他环境事故（如电磁污染，电源故障，设备被盗、被毁等）的破坏。例如，可以对传输介质进行保护，防止物理搭线和窃听；设备上可采用高可用性的硬件、双机多冗余组件；环境上应考虑机房环境及报警系统；操作上应考虑异地备份系统等。

下面介绍应用层、传输层和网络层安全协议和技术。

1. 应用层安全协议

如图 1.16 所示，针对每种应用层协议，可采用相应的安全协议进行安全保护。如针对 SMTP 的 S/MIME、PGP(pretty good privacy)和 PEM(privacy enhanced mail)协议；针对 HTTP 协议的 SHTTP(secure hyper text transfer protocol)协议；针对 DNS 的 DNS Sec、TSIG 和 SKEY 协议；针对 SNMP 的安全协议 SNMPv3 等。这些协议在各自的应用层协议之上进行安全加密、身份鉴别和数据完整性保护。另外，可以使用 Kerberos 协议提供身

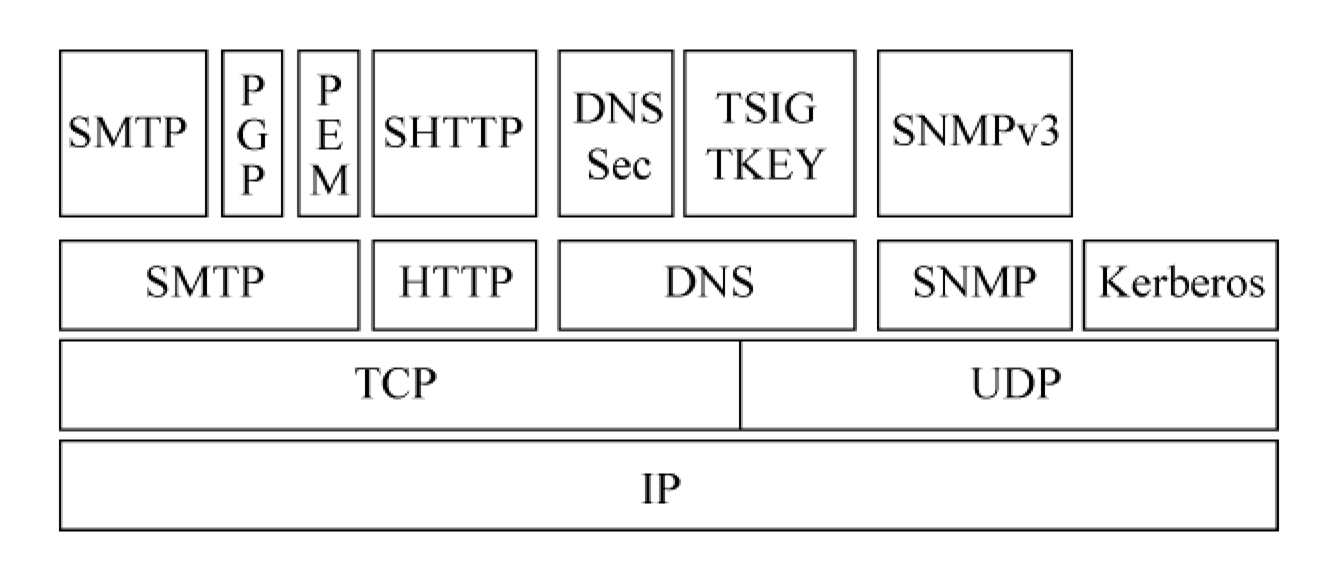


图 1.16　应用层安全协议

份鉴别，它不依赖具体的应用层协议，运行在 UDP 之上。

2. 传输层安全协议

在传输层和应用层之间，可以采用如下安全协议。

- SSL/TLS：安全套接层协议（secure socket layer/transport layer security，SSL/TLS）提供传输协议之上的可靠的端到端安全服务，为两个通信对等实体之间提供保密性、完整性以及鉴别服务。常用于 HTTP 协议，即 HTTPS，也可用于其他应用层协议。
- SSH：SSH（secure shell）是 Internet 工程任务组 IETF（Internet engineering task force）制定的一族协议，目的是在非安全网络上提供安全的远程登录和其他安全服务。SSH 主要解决的是密码在网上明文传输的问题，因此通常用来替代 TELNET、FTP 等协议。SSH 可提供基于主机的认证、机密性和数据完整性服务。
- SOCKS：套接字安全（socket security，SOCKS）是一种网络代理协议。该协议允许使用私有 IP 地址的内部主机通过 SOCKS 服务器访问 Internet，并且连接过程是经过认证的。可提供认证机制、地址解析代理、数据完整性和机密性等服务。

3. 网络层安全协议

IPSec 针对 IP 协议进行数据源鉴别、完整性保护和数据加密。IPSec 包括验证头（authentication header，AH）和封装安全载荷（encapsulation security payload，ESP）两个子协议。其中 AH 对 IP 报文进行认证，同时可保证 IP 数据部分的完整性，但不提供数据加密服务。ESP 主要提供 IP 报文的加密服务，同时提供认证支持，加密过程与具体加密算法相独立。

除了分层的安全协议，还可以采用相应的安全技术来保护网络体系的安全，例如，在网络层安全协议条件下讨论的应用层安全技术如下。

（1）系统扫描：采用系统扫描技术，对系统内部安全弱点进行全面分析，以协助进行安全风险管理。区别于静态的安全策略，系统扫描工具对主机进行预防潜在安全风险的设置。其中包括易猜出的密码、用户权限、文件系统访问权限、服务器设置以及其他含有攻击隐患

的可疑点。

(2) 系统实时入侵检测：为了加强主机的安全，还应采用基于操作系统的入侵检测技术。系统入侵检测技术监控主机的系统事件，实时检查系统的审计记录，从中检测出攻击的可疑特征，并给予响应和处理，如停止入侵进程、切断可疑的通信连接、恢复受损数据等。

(3) 防病毒：防病毒更广泛的定义应该是防范恶意代码，恶意代码不限于病毒，还包括蠕虫、特洛伊木马、逻辑炸弹，以及其他未经同意的软件。防病毒系统应对网关、邮件系统、群件系统、文件服务器和工作站进行全方位的保护，阻断恶意代码传播的所有渠道。这要求防病毒系统对病毒特征码进行及时更新。

(4) 日志和审计：主机的日志和审计记录能够提供有效的入侵检测和事后追查机制。当前应用中的主要网络操作系统（主要包括路由器、交换机、Unix 类和 Windows NT 操作系统等）都能够提供基本的日志记录功能，用于记录用户和进程对于重要文件的更改和对网络资源的访问等。

(5) 用户识别和认证：身份认证技术是实现资源访问控制的重要手段。在应用层或应用系统中可以使用密码、认证令牌（例如智能卡、密码计算器）、基于生物特征的验证等多种手段和方法进行身份鉴别，以防止未授权访问和使用网络资源。

(6) 应用服务器的安全设置：应用层协议在各种应用服务器中实现，为了更好地保护应用系统，必须对应用服务器进行合理设置和安全防护。例如对各级目录的权限进行严格控制；对用户进行授权和管理，对应用服务器软件及时升级和更新等。

网络层可采用如下安全技术进行防护。

(1) 安全的网络拓扑结构：保证网络安全的首要问题就是要合理规划网络拓扑结构，利用网络中间设备的安全机制控制各网络间的访问。

(2) 传输加密：由于入侵者可能窃听机密信息、篡改数据，为了防范这类安全风险，传输系统必须保证数据的机密性与完整性，并且提供抗流量分析的能力。可以选用 IPSec 加密等来满足数据机密性要求。

(3) 网络层漏洞扫描：解决网络层安全问题，首先要清楚网络中存在哪些安全隐患、脆弱点。面对大型网络的复杂性和不断变化的情况，依靠网络管理员的技术和经验寻找安全漏洞并做出风险评估显然是不现实的。解决的方案是，寻找一种能寻找网络安全漏洞、评估并提出修改建议的网络安全扫描工具。

网络漏洞扫描与安全评估系统通过对附属在网络中的设备进行网络安全弱点检测与分析，能够发现并试图修复安全漏洞。

(4) 防火墙：可以采用防火墙进行包过滤，防止非法数据的进入。防火墙是解决子网的边界安全问题、实现网络访问控制的有效方法。防火墙的目的是在内部、外部两个网络之间建立一个安全控制点，通过允许、拒绝或重新定向经过防火墙的数据流，实现对进、出内部网络的服务、访问的审计和控制。

1.5 网络安全评估规范

信息安全评估标准是信息安全评估的行动指南。可信计算机系统安全评估标准(TCSEC)由美国国防部于1985年公布,是计算机系统信息安全评估的第一个正式标准。它把计算机系统的安全分为4类、7个级别,对用户登录、授权管理、访问控制、审计跟踪、隐蔽通道分析、可信通道建立、安全检测、生命周期保障、文档写作和用户指南等内容提出了规范性要求。

信息技术安全评估标准(ITSEC,欧洲白皮书)是由法、英、荷和德欧洲4国在20世纪90年代初联合发布的,它提出了信息安全的机密性、完整性、可用性的安全属性。ITSEC把可信计算机的概念提高到可信信息技术的高度,对国际信息安全的研究、实施产生了深刻的影响。

信息技术安全评价的通用标准(common criteria,CC)是由6个国家(美、加、英、法、德、荷)于1996年联合提出的,并逐渐形成国际标准ISO 15408(《信息技术—安全技术—IT安全评估准则》),包含三个部分:第一部分为“介绍和一般模型”;第二部分为“安全功能需求”;第三部分为“安全认证需求”。该标准定义了评价信息技术产品和系统安全性的基本准则,提出了目前国际上公认的表述信息技术安全性的结构,即把安全要求分为规范产品和系统安全行为的功能要求以及解决如何正确有效地实施这些功能的保障要求。CC标准的发布对信息安全评估具有重要意义,是信息技术安全评价标准以及信息安全技术发展的一个重要里程碑。

ISO 13335(《信息技术安全管理方针》)给出了关于IT安全的保密性、完整性、可用性、审计性、认证性和可靠性6个方面含义,并提出了以风险为核心的安全模型:企业的资产面临很多威胁(包括来自内部的威胁和来自外部的威胁);威胁利用信息系统存在的各种漏洞(如物理环境、网络服务、主机系统、应用系统、相关人员和安全策略等),对信息系统进行渗透和攻击。如果渗透和攻击成功,将导致企业资产的暴露;资产的暴露(如系统高级管理人员由于不小心而导致重要机密信息的泄露),会对资产的价值产生影响(包括直接和间接的影响);风险就是威胁利用漏洞使资产暴露而产生的影响的大小,它可以由资产的重要性和价值所决定;对企业信息系统安全风险的分析,得出了系统的防护需求;根据防护需求的不同制定系统的安全解决方案,选择适当的防护措施,进而降低安全风险,并抗击威胁。该模型阐述了信息安全评估的思路,对企业的信息安全评估工作具有指导意义。

ISO 27001(《信息安全管理基础》)是根据英国的工业、政府和商业的共同需求而开发的一个标准,它分两部分:第一部分为“信息安全管理事务准则”;第二部分为“信息安全管理系统的规范”。目前此标准已经被很多国家采用,并成为国际标准ISO 17799。ISO 17799包含10个控制大项、36个控制目标和127个控制措施。ISO 17799主要提供了有效实施信

息系统风险管理的建议，并介绍了风险管理的方法和过程。企业可以参照该标准制定自己的安全策略和风险评估实施步骤。

AS/NZS 4360：1999 是澳大利亚和新西兰联合开发的风险管理标准，第一版于 1995 年发布。在 AS/NZS 4360：1999 中，风险管理分为建立环境、风险识别、风险分析、风险评价、风险处置、风险监控与回顾、通信和咨询 7 个步骤。AS/NZS 4360：1999 是风险管理的通用指南，它给出了一整套风险管理的流程，对信息安全风险评估具有指导作用。

OCTAVE(operationally critical threat，asset and vulnerability evaluation)是可操作的关键威胁、资产和弱点评估方法和流程。OCTAVE 首先强调的是 O—可操作性，其次是 C—关键系统。OCTAVE 将信息安全风险评估过程划分为三个阶段：阶段一为建立基于资产的威胁配置文件；阶段二为标识基础结构的弱点；阶段三为确定安全策略和计划。

信息保障技术框架(information assurance technical framework，IATF)是美国国家安全局 NSA 的研究成果，它是不断更新、修改的，反映了 NSA 研究信息保障的阶段性成果。目前 IATF 在美国得到了广泛的应用。IATF 基于深度防御战略 Defense-in-Depth，即信息保障依赖人、操作、技术三个因素实现组织的任务与业务运作。IATF 的观点是通过有效结合当前已有成熟技术，充分考虑人员、技术、操作三方面的影响，并衡量防护能力、防护性能、防护耗费和易操作性等各方面因素，得到系统防护的最有效的实用方案，从而实现信息保障。稳健的信息保障状态意味着信息保障的政策、步骤、技术与机制在整个组织的信息基础设施的所有层面上均得以有效实施。IATF 定义了对一个系统进行信息保障的过程，以及该系统中硬件和软件部件的安全要求。遵循这些要求和选择原则，可以对信息基础设施进行深度防御的多层防护。

我国在风险评估方面，主要采用等同国际标准，如 GB 18336 等同采用 ISO 15408 标准。公安部主持制定了中华人民共和国国家标准 GB 17895—1999(《计算机信息系统安全保护等级划分准则》)。该准则将信息系统安全分为 5 个等级：自主保护级、系统审计保护级、安全标记保护级、结构化保护级和访问验证保护级。主要的安全考核指标有身份认证、自主访问控制、数据完整性和审计等，这些指标涵盖了不同级别的安全要求。

1.5.1　可信计算机系统评估准则

在 TCSEC 中，美国国防部按照处理信息的等级和应采用的相应措施，将计算机安全从高到低分为 A、B、C、D 4 类，8 个级别，共 27 条评估准则。随着安全等级的提高，系统的可信度随之增加，风险逐渐减少。

TCSEC 包括 4 个安全等级：

- 无保护级(D 类)。
- 自主保护级(C 类)。
- 强制保护级(B 类)。

• 验证保护级(A类)。

其中,D类是无保护级,为最低的安全保护等级,是为那些经过评估,但不满足较高评估等级要求的系统设计的,这种系统不能满足多用户环境下处理敏感信息的要求。

C类是自主保护级,具有一定的保护能力,采用的措施是自主访问控制和审计跟踪,一般只适用于具有一定等级的多用户环境,具有对主体责任及其动作审计的能力。C类分为C1和C2两个级别:自主安全保护级(C1级)和控制访问保护级(C2级)。C1级的可信计算基(trusted computing base,TCB),即系统中所有保护机构的总称,通过隔离用户与数据,使用户具备自主安全保护的能力。它具有多种形式的控制能力,对用户实施访问控制,为用户提供可行的手段,保护用户和用户组信息,避免其他用户对数据的非法读写与破坏。C1级的系统适用于处理同一敏感级别数据的多用户环境。C2级计算机系统比C1级具有更细粒度的自主访问控制。C2级通过注册过程控制、审计安全相关事件以及资源隔离,使单个用户为其行为负责。

B类为强制保护级,要求是TCB应维护完整的安全标记,并在此基础上执行一系列强制访问控制规则,B类系统中的主要数据结构必须携带敏感标记,系统的开发者还应为TCB提供安全策略模型以及TCB规约,应提供证据证明访问监控器得到了正确的实施。B类分为三个级别:标记安全保护级(B1级)、结构化保护级(B2级)和安全区域保护级(B3级)。

B1级系统要求具有C2级系统的所有特性,在此基础上,还应提供安全策略模型的非形式化描述、数据标记以及命名主体和客体的强制访问控制并消除测试中发现的所有缺陷。在B2级系统中,TCB建立于一个明确定义并文档化、形式化安全策略模型之上,要求将B1级系统中建立的自主和强制访问控制扩展到所有的主体与客体。在此基础上,应对隐蔽信道进行分析。TCB应结构化为关键保护元素和非关键保护元素,鉴别机制应得到加强,提供可信设施管理以支持系统管理员和操作员的职能,提供严格的配置管理控制。B2级系统应具备相当的抗渗透能力。B3级系统支持安全管理员职能、扩充审计机制,当发生与安全相关的事件时,发出信号并提供系统恢复机制。

A类为验证保护级。A类的特点是使用形式化的安全验证方法,保证系统的自主和强制安全控制措施能够有效地保护系统中存储和处理的敏感信息,为证明TCB满足设计、开发及实现等各个方面的安全要求,系统应提供丰富的文档信息。A类分为两个级别:验证设计级A1级和超A1级。安全策略的形式化模型必须得到明确标识并文档化,提供该模型与其公理一致以及能够对安全策略提供足够支持的数学证明,应提供形式化的高层规约,包括TCB功能的抽象定义、用于隔离执行域的硬件、固件机制的抽象定义。

A1级系统要求更严格的配置管理,要求建立系统安全分发的程序,支持系统安全管理员的职能。超A1级在A1级基础上增加的许多安全措施超出了目前的技术发展。随着更多、更好的分析技术的出现,本级系统的要求才会变得更加明确。形式化的验证方法将应用到源码一级,并且时间隐蔽信道将得到全面的分析。在这一级,设计环境变得更重要,形式化高层规约的分析将对测试提供帮助。TCB开发中使用的工具的正确性及TCB运行的软

硬件功能的正确性将得到更多的关注。超 A1 级系统涉及的范围包括系统体系结构、安全测试、形式化规约与验证以及可信设计环境等。

1.5.2　通用准则

CC 定义了一套能满足各种需求的 IT 安全准则，共分为三部分：第一部分为“简介和一般模型”；第二部分为“安全功能要求”；第三部分为“安全保证要求”。其中心内容是：当在安全保护框架(protect profile，PP)和安全目标(security target，ST)中描述评测对象(target of evaluation，TOE)的安全要求时，应尽可能使用其与第二部分描述的安全功能组件和第三部分描述的安全保证组件相一致。

CC 在第一部分描述了对 PP 和 ST 的要求。与传统的软件系统设计相比较，PP 实际上就是安全需求的完整表示，ST 则是通常所说的安全方案。CC 在第二部分和第三部分分别详细介绍了为实现 PP 和 ST 所需要的安全功能要求和安全保障要求，并对安全保证要求进行了等级划分(共分为 7 个等级)。对于安全功能要求，CC 虽然没有进行明确的等级划分，但是在对每一类功能进行具体描述时，要求上还是有差别的。

CC 在对安全保护框架和安全目标的一般模型进行介绍以后，分别从安全功能和安全保证两方面对 IT 安全技术的要求进行了详细描述，主要内容如下。

1. 安全功能要求

CC 将安全功能要求分为以下 11 类：

- 安全审计类。
- 通信类(主要是身份真实性和抗抵赖)。
- 密码支持类。
- 用户数据保护类。
- 标识和鉴别类。
- 安全管理类(与 TSF 有关的管理)。
- 隐秘类(保护用户隐私)。
- TSF 保护类(TOE 自身安全保护)。
- 资源利用类(从资源管理角度确保 TSF 安全)。
- TOE 访问类(从对 TOE 的访问控制确保安全性)。
- 可信路径、信道类。

这些安全类又分为族，族中又分为组件。组件是对具体安全要求的描述，每一个族中的具体安全要求也是有差别的，但 CC 没有以这些差别作为划分安全等级的依据。

对 CC 的 11 个安全类的内容分析可知，其中的前 7 类的安全功能是提供给信息系统使用的，而后 4 类安全功能是为确保安全功能模块的自身安全而设置的。

2. 安全保证要求

具体的安全保证要求分为以下 8 类：

- 配置管理类。
- 分发和操作类。
- 开发类。
- 指导性文档类。
- 生命周期支持类。
- 测试类。
- 脆弱性评定类。
- 保证的维护类。

按照对上述 8 类安全保证要求的不断递增，CC 将 TOE 分为如下 7 个安全保证级。

- 第 1 级：功能测试级。
- 第 2 级：结构测试级。
- 第 3 级：系统测试和检查级。
- 第 4 级：系统设计、测试和复查级。
- 第 5 级：半形式化设计和测试级。
- 第 6 级：半形式化验证的设计和测试级。
- 第 7 级：形式化验证的设计和测试级。

CC 适用于硬件、固件和软件实现的信息技术安全措施，而某些内容因涉及特殊专业技术或仅是信息技术安全的外围技术不在 CC 的范围内。

使用通用评估方法学可以提供评估结果的可重复性和客观性。许多评估准则需要使用专家判断和一定的背景知识。为了增强评估结果的一致性，最终的评估结果应提交给一个认证过程，该过程是一个针对评估结果的独立的检查过程，并生成最终的证书或正式批文。图 1.17 给出了通用准则的评估过程。

1.5.3 信息安全保障技术框架

信息安全保障技术框架（IATF）为保护政府、企业信息及信息基础设施提供了技术指南。IATF 对信息安全保障技术 4 个领域的划分同样适用于信息系统的安全评估，它给出了一种实现系统安全要素和安全服务的层次结构。

信息安全保障技术框架将计算机信息系统划分为本地计算环境、区域边界、网络和基础设施、支撑基础设施 4 个部分。

本地计算环境一般包括服务器、客户端及其上的应用（如打印服务、目录服务等）、操作系统、数据库和基于主机的监控组件（如病毒检测、入侵检测）。

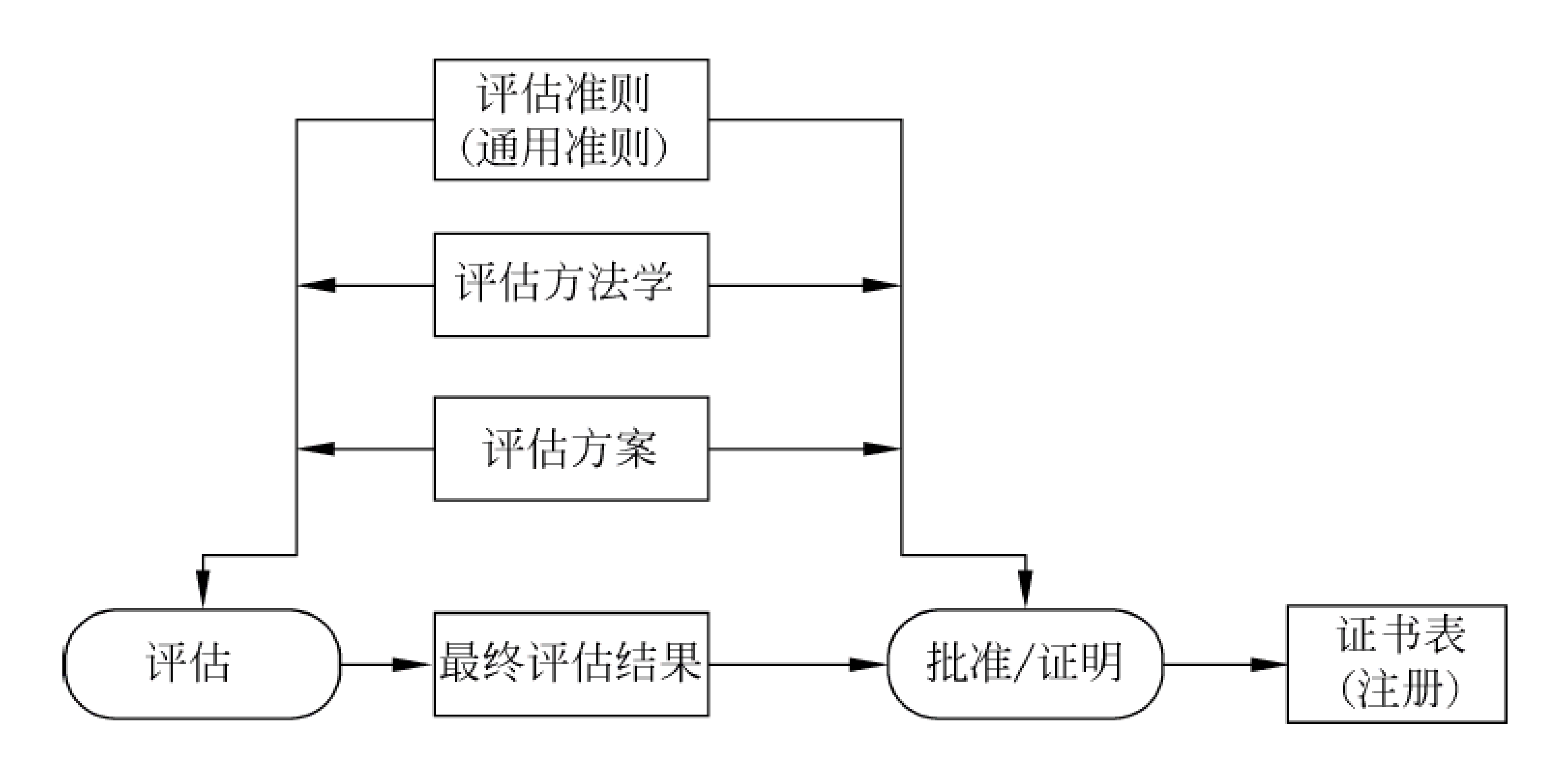


图 1.17 通用准则的评估过程

区域是指在单一安全策略管理下,通过网络连接起来的计算设备的集合。区域边界是区域与外部网络发生信息交换的部分。区域边界确保进入的信息不会影响区域内资源的安全,而离开的信息是经过合法授权的。区域边界上有效的控制措施包括防火墙、VPN、标识和鉴别、访问控制等。监测措施包括基于网络的入侵检测系统(intrusion detection systems,IDS)、脆弱性扫描器、局域网上的病毒检测器等。边界的主要作用是防止外来攻击,它也可以用来对付某些恶意的内部人员。这些内部人员有可能利用边界环境发起攻击,通过开放后门或隐蔽信道来为外部攻击提供方便。

网络和基础设施在区域之间提供连接,包括局域网、园区网、城域网和广域网等。其中包括在网络节点间(如路由器和交换机)传递信息的传输部件(如卫星、微波和光纤等),以及其他重要的网络基础设施组件如网络管理组件、域名服务器及目录服务组件等。对网络和基础设施的安全要求主要是鉴别、访问控制、机密性、完整性、抗抵赖性和可用性。

支撑基础设施提供了在网络、区域及计算环境内进行安全管理、安全服务所使用的基础设施。主要为以下内容提供安全服务:终端用户工作站、WWW 服务、文件、DNS 服务和目录服务等。

IATF 中涉及到两个方面的支撑基础设施:KMI/PKI(key management infrastructure/public key infrastructure)、检测响应基础设施。KMI/PKI 提供了一个公钥证书及传统对称密钥的产生、分发及管理的统一过程。检测及响应基础设施提供针对入侵的检测和响应,包括入侵检测、监控软件等。

使用多层信息安全保障技术来保证信息的安全意味着通过对关键部位提供适当层次的保护就可以为组织提供有效的保护。这种分层的策略允许在恰当的部位存在低保证级别的应用,而在关键部位,如网络边界部分采用高保证级别的应用。区域边界保护内部的计算环境,控制外部用户的非授权访问,同时控制内部恶意用户从区域内发起攻击。

在对信息系统进行安全评估时,可以依据这种多层的深度保卫战略对系统的构成进行合理分析,根据系统面临的各种威胁及实际安全需求分别对计算环境、区域边界、网络和基

础设施、支撑基础设施进行安全评估、对系统的安全保护等级作出评估。在网络上，有三种不同的通信流：用户通信流、控制通信流和管理通信流。信息系统应保证局域内这些通信流的安全。

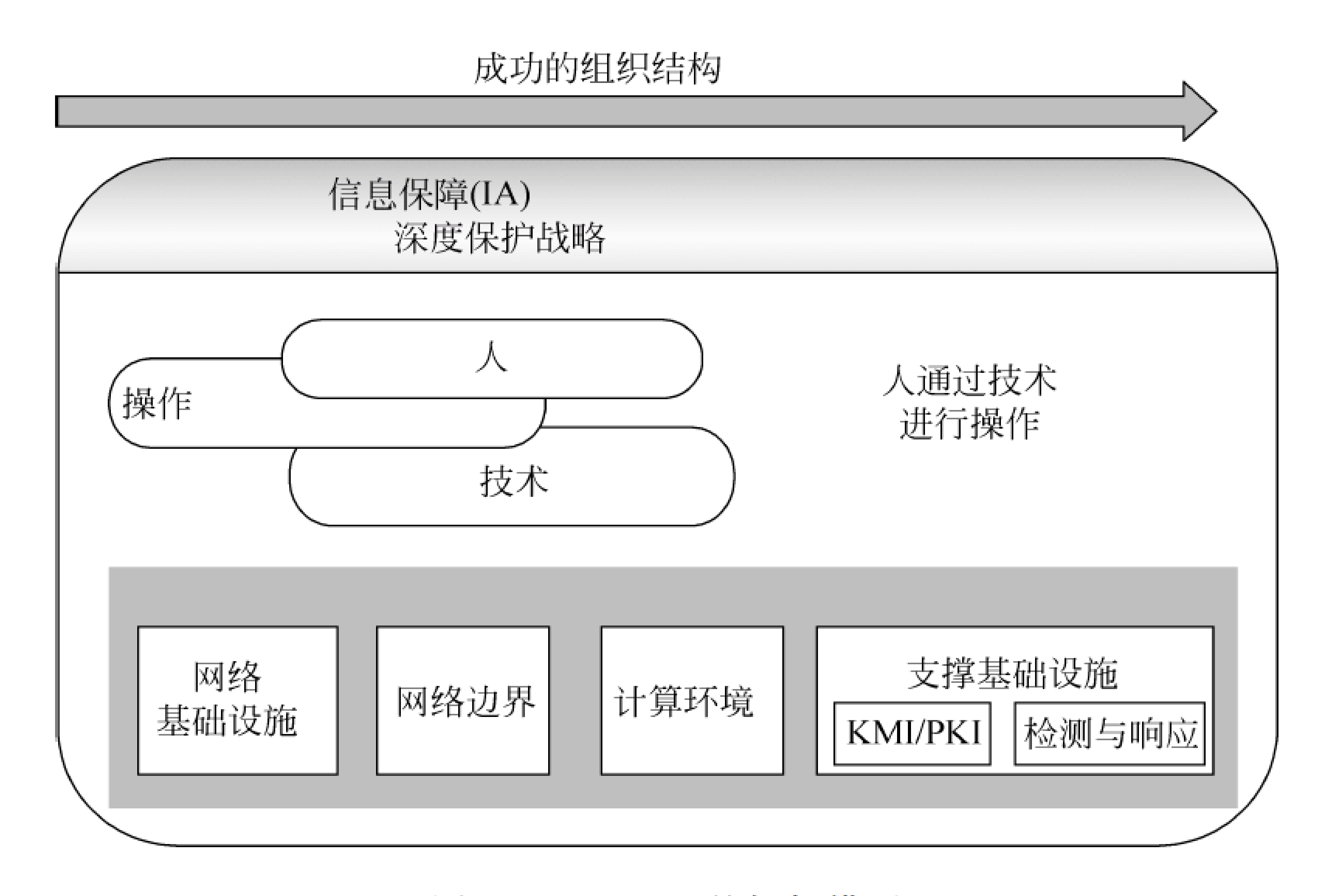


图 1.18　IATF 的框架模型

如图 1.18 所示，IATF 规划的信息安全保障体系包含如下三个要素。

- 人：是信息体系的主体，是信息系统的拥有者、管理者和使用者，是信息保障体系的核心，是第一位的要素，同时也是最脆弱的。正是基于这样的认识，安全管理在安全保障体系中就愈显重要，可以这么说，信息安全保障体系实质上就是一个安全管理的体系，其中包括意识培训、组织管理、技术管理和操作管理等多个方面。
- 技术：是实现信息保障的重要手段，信息保障体系所应具备的各项安全服务就是通过技术机制来实现的。这些技术不单是以防护为主的静态技术体系，还包括防护、检测、响应、恢复并重的动态的技术体系。
- 操作：或称运行，它构成了安全保障的主动防御体系，操作和流程是将各方面技术紧密结合在一起的主动的过程，其中包括风险评估、安全监控、安全审计、跟踪告警、入侵检测和响应恢复等内容。

在明确了信息保障的三项要素之后，IATF 定义了实现信息保障目标的工程过程和信息系统各个方面的安全需求。在此基础上，对信息基础设施就可以做到多层防护，这样的防护被称为“深度防护战略(defense-in-depth strategy)”。

在关于实现信息保障目标的过程和方法上，IATF 论述了系统工程、系统采购、风险管理、认证和鉴别以及生命周期支持等过程，对这些与信息系统安全工程活动相关的方法学作了说明。

信息保障体系即安全循环体系和完整的实施体系，最终实现对于信息和信息系统持续

安全性的保证。这个保障体系包括风险分析、安全防护、安全检测、安全测试与评估、应急响应、恢复和实施体系。

- 风险分析：包括确定系统资源清单，进行脆弱性评估，分析系统风险级别等。风险分析是了解信息系统各方面状态的关键步骤。
- 安全防护：采用相关安全技术、安全机制、安全产品，实现安全防护方案。安全防护是保障信息安全性的重要静态措施。
- 安全检测：使用实时监控、入侵检测和漏洞扫描等技术，对系统进行安全检测，形成资源数据库。
- 测试与评估：定期对系统的安全机制、安全产品和安全状态进行测试和评估，及时发现其存在的安全脆弱性，并进行公正的评估。测试评估与安全检测是保障信息安全性的关键动态措施。
- 应急响应：对突发事件进行快速反应，尽可能减少突发事件对系统的影响，保证系统安全的最小资源集合可用。
- 恢复：当系统遭受破坏时，根据评估系统损失情况，在最短时间内恢复系统数据和系统服务，使系统迅速恢复基本的服务并重建。应急响应能力和恢复能力是信息系统生存性、抗毁性的重要衡量标准。

1.5.4　计算机信息系统安全保护等级划分准则

公安部制定的《计算机信息系统安全保护等级划分准则》(以下简称《准则》)，为计算机信息系统安全法规和配套标准的制定和执法部门的监督检查提供了依据，为安全产品的研制提供了技术支持，为安全系统的建设和管理提供了技术指导，是我国计算机信息系统安全保护等级工作的基础。其相关技术指导资料包括：GA 388—2002(《计算机信息系统安全等级保护操作系统技术要求》)、GA 391—2002(《计算机信息系统安全等级保护管理要求》)、GA/T 387—2002(《计算机信息系统安全等级保护网络技术要求》)、GA/T 389—2002(《计算机信息系统安全等级保护数据库管理系统技术要求》)和 GA/T 390—2002(《计算机信息系统安全等级保护通用技术要求》)等。

《准则》规定了计算机系统安全保护能力的 5 个等级，即第一级为用户自主保护级，第二级为系统审计保护级，第三级为安全标记保护级，第四级为结构化保护级，第五级为访问验证保护级。

用户自主保护级是指计算机信息系统可信计算基通过隔离用户与数据，使用户具备自主安全保护的能力。它具有多种形式的控制能力，对用户实施访问控制，即为用户提供可行的手段，保护用户和用户组信息，避免其他用户对数据的非法读写与破坏。

与用户自主保护级相比，系统审计保护级的可信计算基实施了粒度更细的自主访问控制。它通过登录规程、审计安全性相关事件和隔离资源等手段，使用户对自己的行为负责。

安全标记保护级中，可信计算基具有系统审计保护级的所有功能。此外，还提供有关安全策略模型、数据标记以及主体对客体强制访问控制的非形式化描述，具有准确地标记输出信息的能力，消除通过测试发现的错误。

结构化保护级中，可信计算基建立于一个明确定义的形式化安全策略模型之上，要求将第三级系统中的自主和强制访问控制扩展到所有主体与客体。此外，还要考虑隐蔽信道，可信计算基必须结构化为关键保护元素和非关键保护元素，可信计算基的接口也必须明确定义，使其设计与实现能够经受更充分的测试和更完整的复审，加强了鉴别机制，支持系统管理员和操作员的职能，提供可信设施管理，增强了配置管理控制，系统具有相当的抗渗透能力。

访问验证保护级中，计算机信息系统的可信计算基 TCB 满足访问监控器需求，访问监控器仲裁主体对客体的全部访问。访问监控器本身是抗篡改的；必须足够小，能够分析和测试，支持安全管理员职能，扩充审计机制，当发生与安全相关的事件时发出信号，提供系统恢复机制，系统具有很高的抗渗透能力。

此外，我国还按照安全性评估标准，对安全产品或服务的安全性进行评估、评测和认证。评估标准有《信息安全工程质量管理要求》和《信息安全服务评估准则》等。

本章实验

1. 网络端口扫描
2. 木马攻击

思考题

1. 网络安全服务、安全机制和安全策略之间的区别和联系是什么？
2. 设计并编程实现一个端口扫描程序。
3. TCSEC 和通用准则的主要区别是什么？
4. TCP 会话劫持的原理和实现过程是怎样的？

第 2 章

密码学基础

2.1 密码学概述

密码学(cryptography)是一种将信息表述为不可读的方式,并且使用一种秘密的方法将信息恢复出来的科学。密码学提供的最基本的服务是数据机密性服务,就是使通信者之间可以互相发送信息,并且避免他人读取信息的内容。此外,密码学还可提供数据完整性校验和身份认证(鉴别)等服务。

数据机密性服务的过程如图 2.1 所示。原始信息称为明文(plain text),明文经过加密后形成密文(cipher text),密文经过解密后恢复成原始明文。

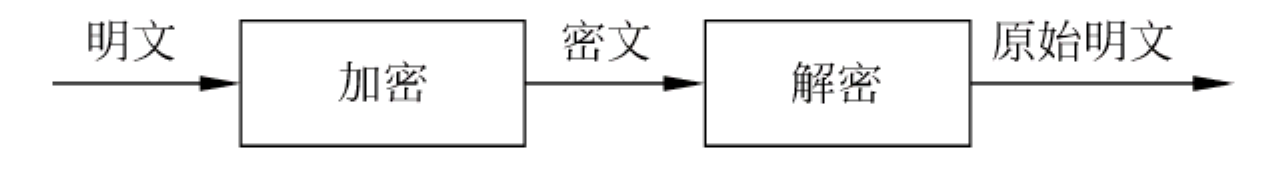


图 2.1　数据机密性服务的过程

密码学包括两个分支,即密码编码学和密码分析学。前者研究各种加密方案,称为密码体制(cryptosystem)或密码(cipher)。在不知道加密细节的情况下解密消息密文的技术属于密码分析学研究的范畴。密码学是对这两门分支进行综合分析,系统研究的科学,是保护信息安全的主要手段之一。

最早的密码学应用可追溯到公元前 2000 年古埃及人使用的象形文字。这种文字由复杂的图形组成,其含义只被为数不多的人掌握。而最早将现代密码学概念运用于实际的人是恺撒大帝。他不太相信负责他和他手下将领通信的传令官,因此他发明了一种简单的加密算法把他的信件加密。

第二次世界大战以后,由于与计算机技术的结合,密码学的理论研究与实际应用得到了飞速的发展,随之产生了许多新的分支理论,如微粒照片、数字图片水印技术和其他很多隐藏被传递和存储的信息的方法。其中,最常见的是利用计算机将明文和密码变成密文和将

密码和密文变成明文。

2.1.1 密码算法和密钥

如图 2.2 所示,密码系统(cryptography)由密码算法(cryptographic algorithm)以及所有可能的明文、密文和密钥(key)组成。密码算法是进行加密或解密的数学函数。密码算法在通常情况下有两个相关的函数,一个用来加密,一个用来解密。密钥是密码算法的输入参数,是一对用来加密和解密的字符串。

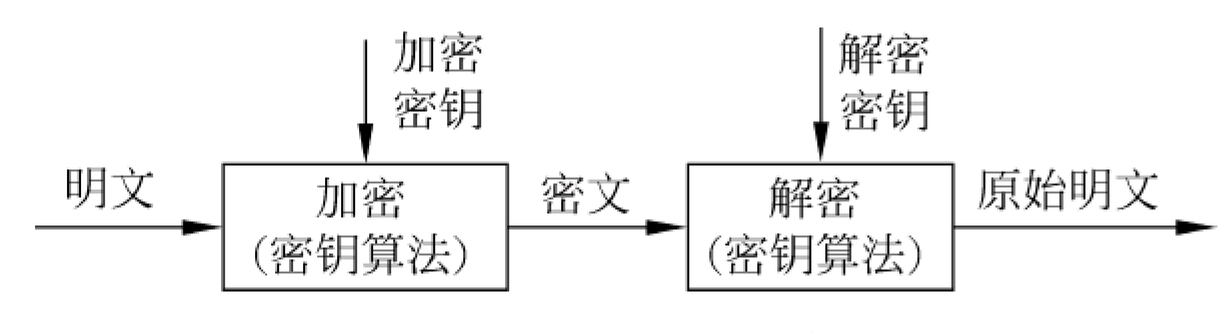


图 2.2 密码系统的构成

密码体制中,如果数据的保密性基于密码算法的秘密性,这种算法称为受限制的算法。受限制的算法不能进行质量控制或标准化,每个用户和组织需确定他们自己唯一的算法。这种受限制的加密算法通常被应用于军用密码系统中。

现代的商用密码系统中,加密后数据的安全性一般是基于密钥的安全性,而不是基于算法的安全性,即密码算法可以公开,也可以被分析,只要密钥不泄露,在现有条件和有效时间内,加密数据就无法被破解。

2.1.2 密码算法分类

密码算法有三种:对称密钥算法、公开密钥算法和哈希算法。

1. 对称密钥算法

对称密码体制采用对称密钥算法(symmetric algorithm)实现。对称密钥算法又称单密钥算法,或秘密密钥算法。对称密钥算法的特点是加密密钥能够从解密密钥推算出来,反之亦然。

在大多数对称密钥算法中,加密和解密采用同一对密钥。它要求发送者和接收者在安全通信之前,商定一个密钥。对称密码的安全性依赖于密钥,密钥必须保密。

对称密钥算法可以提供数据机密性服务、身份认证和数据完整性服务。其特点是加、解密速度快。

2. 公开密钥算法

公开密钥算法又称双密钥算法,或非对称密钥算法。在公钥密码体制中,加密和解密使用不同的密钥:加密密钥可公开,称为公钥(public key);解密密钥需保密,称为私钥(private

key)。即使用＜公钥,私钥＞对信息进行加密解密,使用公钥加密,私钥解密。任何人均可知晓公钥并可以利用公钥加密信息,但只有使用相应的私钥才能解密由该公钥加密的信息。

公开密码体制对密码算法的要求是由公钥不能推导出私钥,并且从一段明文和相应的密文中难以破解私钥。公钥算法较对称密钥算法的优势是它克服了分发密钥带来的不安全性。其缺点是加、解密速度相对于对称密钥算法要慢。

公钥密码体制的基础是单向陷门函数(trapdoor one-way function),其特性是:除非知道某种附加的信息,否则这样的函数在一个方向上容易计算,而在相反的方向上计算不可行。这种附加信息就是私钥,即除非知道私钥,否则只能加密,解密非常困难。

公钥算法可以提供数据机密性服务、身份认证和数字签名服务。

下面给出利用公钥密码进行安全通信的过程示例。

① Alice 和 Bob 选用一个公钥密码系统。

② 系统为 Bob 生成一对密钥＜E,D＞,其中 E 为 Bob 的公钥,D 为对应的私钥。

③ Bob 将他的公钥 E 传递给 Alice。

④ Alice 使用 Bob 的公钥 E 加密消息,然后将加密的消息传递给 Bob。

⑤ Bob 使用他的私钥 D 解密 Alice 的消息。

3. 哈希算法

哈希(hash)算法又称消息摘要(message digest)函数、杂凑算法、单向散列函数,其特性是可以由任意长的消息计算出一个定长的字符串。该字符串称为“哈希”、“散列值”或“消息摘要”。设哈希函数为 $h()$,消息 m 的哈希为 $h(m)$,该算法具有以下特点:

- $h()$能用于任意大小的分组。
- $h()$能产生定长的输出。
- 对于任何给定的消息 m,$h(m)$要容易计算,使得硬件和软件的实现成为实际可能。
- 对于任何给定的散列值 x,寻找 m,使得 $h(m)=x$,这在计算上不可行。即哈希函数具有单向性,无法从消息的散列值中恢复出原消息。
- 对于任何给定的分组 x,寻找不等于 x 的分组 y,使得 $h(x)=h(y)$,这在计算上不可行。即哈希函数具有弱抗冲突性,难以找到两个相等的消息,其散列值相同。
- 寻找任何一对(x,y),使得 $h(x)=h(y)$,这在计算上不可行。即哈希函数具有强抗冲突性。

哈希函数的这些特性使得其可被广泛应用于密码系统、消息完整性服务、消息指纹以及数字签名中。

在密码系统中,可以将密码经过哈希函数计算得到密码的哈希值,保存在系统中,然后对于用户输入的密码再次计算其哈希值,通过比较这两个哈希值可以判断用户输入的密码是否正确。由于哈希函数具有单向性,攻击者即使获得了密码的哈希值,也难以破解密码。

由于哈希函数具有弱抗冲突性,消息的一位发生变化会导致其散列值发生变化,因此哈

希函数可用于消息完整性服务,用来防止消息在传输过程中被篡改。例如可被用于数字签名和产生消息指纹。

数字签名的应用将在下一章详细讨论,下面给出消息指纹的应用示例。在一个应用系统中,如果要知道某些大型数据(或程序)每天是否被修改,可以定时保存这些数据的一个副本,然后把它和当前的副本进行比较。这个副本可以是该数据的哈希值,哈希值是定长的,因此可以大大减少所需的存储空间。如果哈希值没有改变,则可以判断数据没有被修改。

2.1.3 密码分析与计算复杂性

密码分析是在不知道密钥的情况下,恢复出明文的科学,密码分析可以发现密码体制的弱点。针对密码算法的攻击包括唯密文攻击、已知明文攻击和选择明文攻击三种基本形式。

(1) 唯密文攻击(ciphertext-only attack)

密码分析者拥有一些密文,并试图通过分析密文破译密码。有多种方法可以在获得密文的情况下破解出明文。例如:攻击者可以搜索所有可能的密钥,然后使用每个密钥逐个对密文进行解密。前提之一是攻击者能够区分正确的明文和错误的解密结果,因此这种攻击方法也称为可识别明文攻击。此外,攻击者还必须获得足够多的密文,如果密文的数量不足,则不足以从中恢复出明文。

(2) 已知明文攻击(known-plaintext attack)

密码分析者不仅可以得到一些消息的密文,而且也知道这些消息的明文。在这种攻击中,攻击者获得了明文以及与之对应的密文。从这些已知明文中,攻击者可以获得一些常用的字母的替换关系。有些密码算法对于唯密文攻击是安全的,但并不能有效抵抗已知明文攻击。这种情况下,如果密码系统中使用了这类算法,则应尽量减少攻击者获得(明文,密文)的可能。

(3) 选择明文攻击(chosen-plaintext attack)

某些情形下,攻击者可以选择任意的明文输入到密码系统中并获得该明文对应的密文。这种攻击成为选择明文攻击。

某些能够抵抗唯密文攻击和已知明文攻击的密码算法可能无法抵抗选择明文攻击。例如,如果攻击者知道系统的明文是有限的序列{T1,T2,…,Tn}中的某个值,则攻击者只需要把每个可能的明文输入系统,获取对应的密文,再对照该有限明文序列,就可以从密文中推算出其对应的明文。

在有效的时间内,密码系统必须能够抵抗上述三种攻击,这样的密码算法和密码体制才被认为是安全的。

在密码体制中,没有绝对安全的密码技术。如果具有无限计算资源的密码分析者无法破解该密码,这样的密码系统被认为具有理论安全性(无条件安全性)。然而实际上密码体制都不是绝对安全的,只是有些密码算法加密的密文在现有条件下不可破解。

2.2 对称密钥算法

2.2.1 DES

数据加密标准(data encryption standard,DES),是由 IBM 公司研制的一种加密算法,美国国家标准局于 1977 年宣布把它作为非机要部门使用的数据加密标准。DES 是一个分组加密算法,以 64 位为分组对数据加密。同时 DES 也是一个对称密钥算法：加密和解密用的是同一个密钥。它的密匙长度是 56 位(每个第 8 位都用作奇偶校验),密匙可以是任意的 56 位的数,而且可以在任意时候改变。其中有极少量的数被认为是弱密匙,但是很容易避开它们。所以保密性依赖于密钥。

如图 2.3 所示,DES 对 64 位的明文分组 M 进行操作,M 经过一个初始置换 IP(initial permutation)置换成 m_0,将 m_0 明文分成左半部分和右半部分,即 $m_0=(L_0,R_0)$,各 32 位长。然后进行 16 轮完全相同的运算,这些运算被称为函数 f,在运算过程中数据与密匙结合。经过 16 轮后,左右半部分合在一起经过一个末置换。

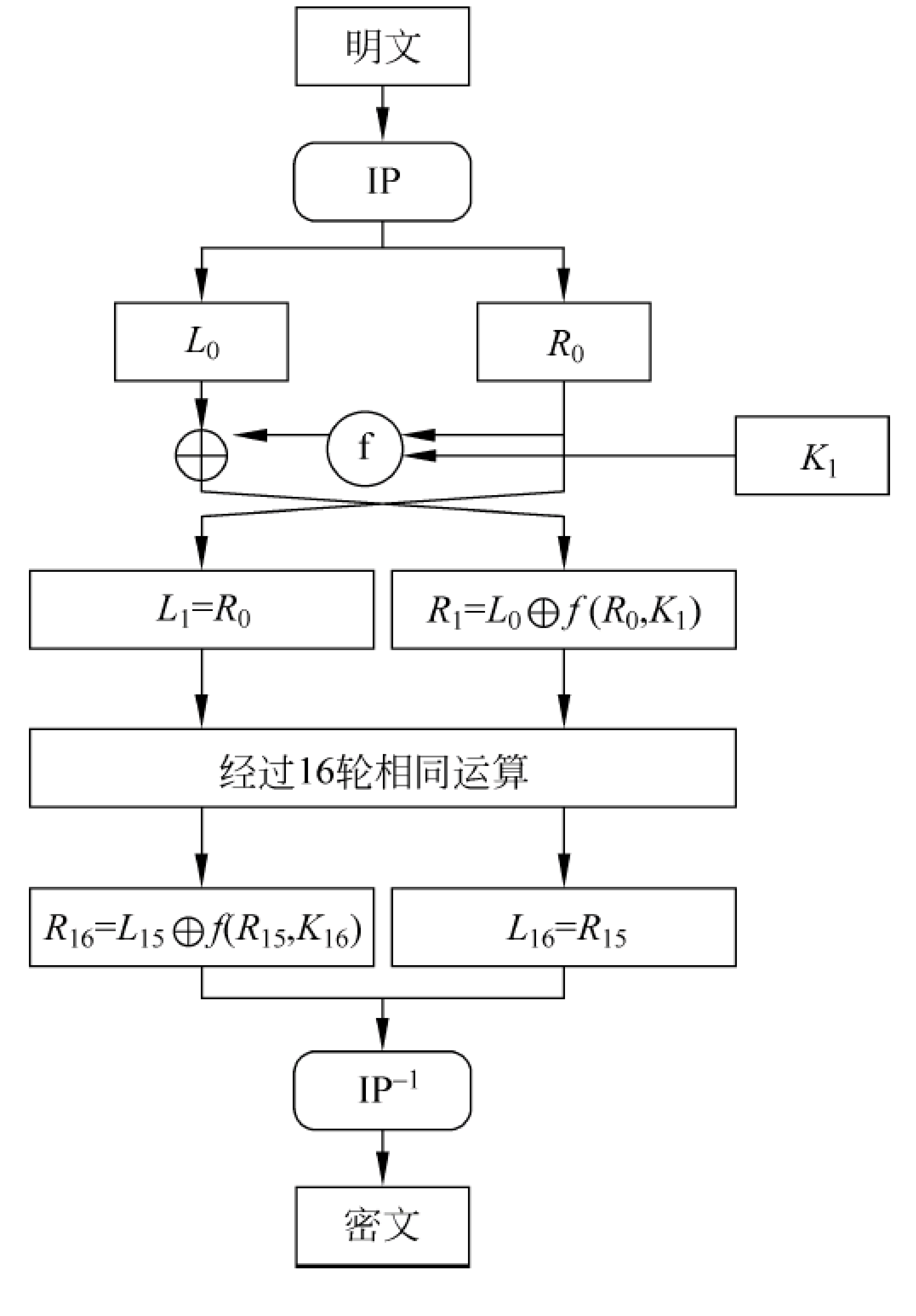


图 2.3 DES 加密算法

在每一轮中,密匙位移位,然后再从密匙的 56 位中选出 48 位。通过一个扩展置换将数据的右半部分扩展成 48 位,并通过一个异或操作替代成新的 32 位数据,再将其置换一次。这 4 步运算构成了函数 f。然后,通过另一个异或运算,函数 f 的输出与左半部分结合,其结果成为新的右半部分,原来的右半部分成为新的左半部分。将该操作重复 16 次,就实现了加密过程。

经过精心选择的各种操作,DES 获得了一个非常有用的性质：加密和解密使用相同的算法。DES 加密和解密唯一的不同是密匙的次序相反。如果各轮加密密匙分别是 $K_1,K_2,K_3,\cdots,K_{16}$那么解密密匙就是 $K_{16},K_{15},K_{14},\cdots,K_1$。

DES 具有如下几种工作模式。

(1) 电子密本模式(electronic codebook,ECB)

将明文分成 n 个 64 位的分组,如果明文长度不是 64 位的倍数,则在明文末尾填充适当

数目的规定符号。对明文组用给定的密钥分别进行加密，再将密文组按顺序连接起来就可以获得密文。

(2) 密文分组链接模式(cipher block chaining，CBC)

在 CBC 模式下，每个明文组在加密之前先与前一个密文组按位模 2 求和后，再对结果运用 DES 加密。对于第一个明文组，由于还没有反馈密文，需预置一个初始向量 IV (initialization vector)。CBC 模式通过反馈使输出密文与以前的各明文相关，从而实现隐蔽明文的目的，但同时也可能引起错误传播的发生。

(3) 密文反馈模式(cipher feedback，CFB)

CFB 模式将 DES 作为一个流密码产生器。首先，将移位寄存器的最右边 64 位送到 DES 进行加密，再将加密结果的最左边 n 位与 n 位明文分组作异或运算得到密文分组。随后，将得到的密文分组送到移位寄存器的最右端，其他位向左移 n 位，最左端 n 位丢弃。然后继续下一分组的运算。对第一个分组同样需要预置初始向量。CFB 是基于密文反馈的，对信道错误较敏感，也可能造成错误传播。

(4) 输出反馈模式(output feedback，OFB)

OFB 模式也将 DES 作为密文流产生器，不同的是它将输出的 n 位密钥直接反馈至移位寄存器，即 DES 的输入端。

2.2.2 3DES

3DES(triple DES)是 DES 向高级加密标准(advanced encryption standard，AES)过渡的一种加密算法(1999 年，NIST 将 3DES 指定为过渡的加密标准)，是 DES 的一个更安全的变形。它以 DES 为基本模块，通过组合分组方法设计出分组加密算法，其具体实现如下：设 $E_{k()}$ 和 $D_{k()}$ 代表 DES 算法的加密和解密过程，K 代表 DES 算法使用的密钥，M 代表明文，C 代表密文，则 3DES 加密、解密过程为

$$\text{加密：} C = E_{k3}(D_{k2}(E_{k1}(M)))$$

$$\text{解密：} M = D_{k1}(E_{k2}(D_{k3}(C)))$$

$K1$、$K2$、$K3$ 决定了算法的安全性，若三个密钥互不相同，本质上就相当于使用一个长度为 168 位的密钥进行加密。多年来，它在对付强力攻击时是比较安全的。若数据对安全性要求不那么高，或者要与原来的 DES 保持兼容，则可以选择 $K1=K2$ 或者 $K2=K3$。

2.2.3 其他对称密钥算法

1. IDEA 算法

IDEA(international data encryption algorithm)即国际数据加密算法，它的原型是 PES

(proposed encryption standard)。对 PES 改进后的新算法称为 IPES,并于 1992 年改名为 IDEA。

IDEA 是一个分组长度为 64 位的分组密码算法,密钥长度为 128 位,同一个算法即可用于加密,也可用于解密。

IDEA 的加密过程包括两部分:

① 输入的 64 位明文组分成 4 个 16 位子分组。4 个子分组作为算法第一轮的输入,共进行 8 轮的迭代运算,产生 64 位的密文输出。

② 输入的 128 位会话密钥产生 8 轮迭代所需的 52 个子密钥(8 轮运算中每轮需要 6 个,还有 4 个用于输出变换)。

IDEA 的解密过程和加密过程相同,只是对子密钥的要求不同。

2. Blowfish 算法

Blowfish 是 Bruce Schneier 设计的,可以免费使用。

Blowfish 是一个 16 轮的分组密码,明文分组长度为 64 位,使用变长密钥(从 32 位到 448 位)。Blowfish 算法由两部分组成:密钥扩展和数据加密。

密钥扩展将密钥转变成 18 个 32 位的子密钥,总共需要 521 次迭代来产生所需的全部子密钥。可以将此扩展密钥存储而无须每次重复计算。

数据加密共进行 16 轮的迭代。解密过程与加密过程完全一样,只是密钥以逆序使用。

3. GOST 算法

GOST 是前苏联设计的分组密码算法,为前苏联国家标准局所采用。

GOST 的消息分组为 64 位,密钥长度为 256 位,采用 32 轮迭代。加密时,首先将输入的 64 位明文分成左半部分 L 和右半部分 R,设第 i 轮的子密钥为 K_i,则 GOST 的第 i 轮为

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

首先右半部分与第 i 轮的子密钥进行模 2^{32} 加和,其结果分成 8 个 4 位分组,每个分组输入不同的 S 盒。S 盒将输入的数字进行置换。然后将 8 个 S 盒的输出重组成 32 位字;接下来将 32 位循环左移 11 位后与上一轮的左半部分进行异或运算得到本轮运算结果的右半部分,而原右半部分作为本轮运算结果的左半部分。至此,一轮运算结束,开始下一轮运算。

GOST 子密钥产生的过程很简单,256 位密钥被划分为 8 个 32 位分组,各轮采用不同的子密钥。解密时,子密钥采用相反的顺序。

GOST 的设计者打算在有效性和安全性之间达到平衡,他们修改了 DES 的基本设计以便产生一个更适宜于软件实现的算法,并通过增大密钥长度、对 S 盒保密、增加加密轮数来增强其安全性。

4. RC5 算法

RC5 分组密码算法是 1994 年由 Ronald L. Rivest 发明的，并由 RSA 实验室分析。它是参数可变的分组密码算法，三个可变的参数是分组长度、密钥长度和加密迭代轮数。RC5 算法包括三部分：密钥扩展、加密算法和解密算法。

Rivest 设计了 RC5 的一种特殊的实现方式，RC5 算法有一个面向字的结构：RC5-w/r/b，这里 w 是字长，其值可以是 16、32 或 64，对于不同的字长，明文和密文块的分组长度为 2w 位，r 是加密轮数，b 是密钥字节长度。

要创建密钥组，RC5 算法加密时将密钥字节复制到 32 位字的数组中，如果需要，最后一个字可以用零填充。然后利用线性同余发生器模 2 初始化数组 S，然后将构造字阵 L 与 S 混合。在创建完密钥组后开始进行对明文的加密，加密时，先将明文分组划分为两个 32 位字：A 和 B（在假设处理器字节顺序是 little-endian、w＝32 的情况下，第一个明文字节放入 A 的最低位，第四个明文字节放入 A 的最高位，第五个明文字节放入 B 的最低位，以此类推，最后一个字节放入 B 的最高位），输出的密文是在寄存器 A 和 B 中的内容。解密也是把密文分组划分为两个字：A 和 B（存储方式和加密一样），进行加密运算的逆运算。

RSA 试验室花费了相当长的时间来分析 64 位分组的 RC5 算法，在 5 轮后统计特性看起来非常好，在 8 轮后，每一个明文位至少影响一个循环，对于 15 轮或以上的 RC5 的差分攻击是失败的。在 6 轮后线性分析就是安全的了。所以 Rivest 推荐至少 12 轮。

2.3 公钥算法

2.3.1 RSA

RSA 是第一个既能用于数据加密也能用于数字签名的算法。它易于理解和操作，也很流行。算法的名字以发明者的名字命名：Ron Rivest，AdiShamir 和 Leonard Adleman。RSA 是被研究得最广泛的公钥算法，经历了各种攻击的考验，逐渐为人们接受，并且被普遍认为是最优秀的公钥方案之一。

RSA 算法描述如下。

① 选取两个大素数 p 和 q（保密）。

② 计算 $n=pq$（公开），$\phi(n)=(p-1)(q-1)$（保密）。

③ 随机选取正整数 e，$1<e<\phi(n)$，满足 $\gcd(e,\phi(n))=1$，e 是公开的加密密钥。

④ 计算 d，满足 $de\equiv 1(\bmod\phi(n))$，d 是保密的解密密钥。

⑤ 加密变换：对明文 $m\in Z_n$，密文为

$$c = n^e \bmod n$$

⑥ 解密变换：对密文 $c \in Z_n$，明文为

$$m = c^d \bmod n$$

RSA 公钥密码体制的安全性是基于大整数的素数分解问题的难解性。

从下面的讨论可以看出，破译 RSA 的方法至少和因子分解问题一样困难。

如果密码分析者能够分解 n 的因子 p 和 q，就可以容易地求出 $\phi(n)$ 和解密密钥 d，从而破译 RSA。因此，破译 RSA 不可能比因子分解问题更困难。

如果密码分析者能够不对 n 进行因子分解而求得 $\phi(n)$，则可以根据

$$\mathrm{de} \equiv 1(\bmod \phi(n))$$

求得解密密钥 d，从而破译 RSA。因为

$$p + q = n - \phi(n) + 1$$

$$p - q = \sqrt{(p+q)^2 - 4n}$$

所以知道 $\phi(n)$ 和 n 就可以很容易地求得 p 和 q，从而成功地分解 n。因此，不对 n 进行因子分解而直接计算 $\phi(n)$ 并不比对 n 进行因子分解更容易。

如果密码分析者能够既不对 n 进行因子分解也不求 $\phi(n)$ 而直接求得解密密钥 d，则他就可以计算 $ed-1$，$ed-1$ 是 $\phi(n)$ 的倍数。而利用 $\phi(n)$ 的倍数可以容易地分解出 n 的因子。因此，直接计算解密密钥 d 并不比对 n 进行因子分解更容易。

为保证 RSA 的安全性，在实际应用中选取的素数 p 和 q 应足够大，n 的长度在 1024 位至 2048 位是比较合理的。此外，为避免选取容易分解的整数 n，研究人员建议对 p 和 q 采取如下限制：

① p 和 q 的长度应该相差不多。

② $p-1$ 和 $q-1$ 都应该包含大的素因子。

③ $\gcd(p-1, q-1)$ 应该很小。

2.3.2　Diffie-Hellman

D-H（Diffie-Hellman）是由 Whitfield Diffie 和 Martin Hellman 在 1976 年公布的一种密钥一致性算法。Diffie-Hellman 是一种建立密钥的方法，而不是加密方法。然而，它所产生的密钥可用于加密、进一步的密钥管理或任何其他的加密方式。

由于该算法本身限于密钥交换的用途，被许多商用产品用作密钥交换技术，因此该算法通常称之为 Diffie-Hellman 密钥交换。这种密钥交换技术的目的在于使得两个用户安全地交换一个秘密密钥以便用于以后的报文加密。

Diffie-Hellman 密钥交换算法的有效性依赖于计算离散对数的难度。

可以如下定义离散对数：首先定义一个素数 p 的原根，为其各次幂产生从 1 到 $p-1$ 的所有整数根，也就是说，如果 a 是素数 p 的一个原根，那么数值 $a \bmod p, a^2 \bmod p, \cdots,$

$a^{p-1} \bmod p$ 是各不相同的整数,并且以某种排列方式组成了从 1 到 $p-1$ 的所有整数。对于一个整数 b 和素数 p 的一个原根 a,可以找到唯一的指数 i,使得 $b = a^i \bmod p$,其中 $0 \leqslant i \leqslant (p-1)$,指数 i 称为 b 的以 a 为基数的模 p 的离散对数。

Diffie-Hellman 密钥交换算法描述如下。

① 有两个全局公开的参数,一个素数 p 和一个整数 a,a 是 p 的一个原根。

② 假设用户 A 和 B 希望交换一个密钥,用户 A 秘密产生一个随机数 X_A,计算 $Y_A = a^{X_A} \bmod p$,然后将 Y_A 发送给用户 B。同样,用户 B 秘密产生一个随机数 X_B,计算 $Y_B = a^{X_B} \bmod p$,然后将 Y_B 发送给用户 A。

③ 用户 A 产生共享秘密密钥的计算方式是 $K=(Y_B)^{X_A} \bmod q$。同样,用户 B 产生共享秘密密钥的计算是 $K=(Y_A)^{X_B} \bmod q$。这两个计算产生相同的结果

$$\begin{aligned} K &= (Y_B)^{X_A} \bmod q = (a^{X_B} \bmod q)^{X_A} \bmod q \\ &= (a^{X_B})^{X_A} \bmod q = a^{X_B X_A} \bmod q \\ &= (a^{X_A})^{X_B} \bmod q = (a^{X_A} \bmod q)^{X_B} \bmod q \\ &= (Y_A)^{X_B} \bmod q \end{aligned}$$

因此相当于双方已经交换了一个相同的秘密密钥。

因为 X_A 和 X_B 是保密的,攻击者可以利用的参数只有 q,a,Y_A 和 Y_B。因而攻击者只能通过取离散对数来确定密钥。

Diffie-Hellman 密钥交换算法的安全性依赖于这样一个事实:虽然计算以一个素数为模的指数相对容易,但计算离散对数却很困难。对于大的素数,计算出离散对数几乎是不可能的。

然而,Diffie-Hellman 密钥交换容易受到中间人攻击,利用数字签名可以抵御中间人攻击。Diffie-Hellman 密钥交换也可以扩展到三方或多方的密钥交换。

2.4 哈希算法

2.4.1 MD5

消息摘要函数 MD5(message-digest algorithm 5)在 20 世纪 90 年代初由 Ronald L. Rivest 开发,经 MD2、MD3 和 MD4 发展而来。它的作用是让大容量信息在用数字签名软件签署私人密匙前被"压缩"成一种保密的格式(就是把一个任意长度的字节串变换成一定长的大整数)。

Rivest 在 1989 年开发出 MD2 算法。在这个算法中,首先对信息进行数据补位,使信息的字节长度是 16 的倍数。然后,以一个 16 位的检验追加到信息末尾,并且根据这个新产生的信息计算出散列值。后来发现如果忽略了检验和将产生 MD2 冲突。

为了加强算法的安全性，Rivest 在 1990 年又开发出 MD4 算法。MD4 算法同样需要填补信息以确保信息的字节长度加上 448 后能被 512 整除。然后，添加一个以 64 位二进制表示的消息长度信息。信息被处理成 512 位迭代结构的区块，每个区块要通过三个不同步骤的处理。Den Boer 和 Bosselaers 以及其他人很快地发现了攻击 MD4 版本中第一步和第三步的漏洞。于是 MD4 就此被淘汰。尽管 MD4 算法在安全上有很大的漏洞，但它对其后开发出来的几种信息安全加密算法却有着不可忽视的引导作用。除了 MD5 以外，其中比较有名的还有 sha-1、ripe-md 和 haval 等。

一年以后，即 1991 年，Rivest 开发出技术上更趋近成熟的 MD5 算法。它在 MD4 的基础上增加了“安全—带子”(safety-belts)的概念。虽然 MD5 比 MD4 稍微慢一些，但却更加安全。这个算法很明显的由 4 个和 MD4 设计有少许不同的步骤组成。在 MD5 算法中，消息摘要的大小和填充的必要条件与 MD4 完全相同。

MD5 的典型应用是对一段消息(message)产生消息摘要(message-digest)，以防止被篡改。比如，在 Unix 下有很多软件在下载的时候都有一个文件名相同，文件扩展名为.MD5 的文件，在这个文件中通常只有一行文本，大致结构如：

MD5(example.tar.gz) = 0ca175b9c0f726a831d895e269332461

这就是 example.tar.gz 文件的数字签名。MD5 将整个文件当作一个大文本信息，通过其不可逆的字符串变换算法，产生了这个文件对应的唯一的 MD5 消息摘要。如果在以后传播这个文件的过程中，无论文件的内容发生了任何形式的改变(包括人为修改或者下载过程中线路不稳定引起的传输错误等)，只要对这个文件重新计算 MD5 时就会发现消息摘要不相同，由此可以确定得到的只是一个不正确的文件。如果再有一个第三方的认证机构，使用 MD5 还可以防止文件作者的“抵赖”，这就是所谓的数字签名应用。

MD5 还被广泛用于加密和解密技术上。比如在 Unix 系统中用户的密码就是以 MD5(或其他类似的算法)经加密后存储在文件系统中的。当用户登录的时候，系统把用户输入的密码计算成 MD5 值，然后再去和保存在文件系统中的 MD5 值进行比较，进而确定输入的密码是否正确。通过这样的步骤，系统在并不知道用户密码的情况下就可以确定用户登录系统的合法性。这不但可以避免用户的密码被具有系统管理员权限的用户知道，而且还在一定程度上增加了密码被破解的难度。

对 MD5 算法简要的描述如下。

首先填充消息，先填充一位 1，再填充一定数量的 0，使其长度为一个比 512 的倍数小 64 位的数。然后将原消息长度的 64 位附加在填充后的消息之后。初始化用于计算消息摘要的 128 位缓冲区，这个缓冲区由 4 个 32 位寄存器组成。按 512 位的分组处理输入消息，每个循环包括 4 轮，每一轮进行 16 次操作。以下是每次操作中用到的 4 个非线性函数(每轮一个)。

$$F(X,Y,Z) = (X\&Y) \mid ((\sim X)\&Z)$$
$$G(X,Y,Z) = (X\&Z) \mid (Y\&(\sim Z))$$

$$H(X,Y,Z) = X^{\wedge} Y^{\wedge} Z$$

$$I(X,Y,Z) = Y^{\wedge} (X \mid (\sim Z))$$

其中，& 表示“与”，| 表示“或”，～表示“非”，^ 表示“异或”。

最后，由 4 个寄存器的输出按低位字节在前的顺序得到 128 位的消息摘要。

2.4.2 SHA

SHA(secure hash algorithm，安全散列算法)是美国国家安全局(NSA)设计，美国国家标准与技术研究院(NIST)发布的一系列密码散列函数。

最初发明的算法于 1993 年发布，称为安全散列标准(secure Hash standard)，FIPS PUB 180。这个版本现在常被称为 SHA-0。它在发布之后很快就被 NSA 撤回，并且以 1995 年发布的修订版本 FIPS PUB 180-1(通常称为 SHA-1)取代。根据 NSA 的说法，它修正了一个在原始算法中会降低密码安全性的错误。然而 NSA 并没有提供任何进一步的解释证明该错误已被修正。1998 年，在一次对 SHA-0 的攻击中发现这次攻击并不能适用于 SHA-1——不知道这是否就是 NSA 所发现的错误，但这或许暗示这次修正已经提升了安全性。

SHA-1 产生消息摘要的过程类似 MD5，输入为长度小于 2^{64} 位的消息，输出为 160 位的消息摘要。具体过程如下。

① 填充消息。首先将消息填充为 512 位的整数倍，填充方法和 MD5 相同：先填充一个 1；然后填充一定数量的 0，使其长度比 512 的倍数少 64 位；接下来附加上原消息长度的 64 位。

② 初始化缓冲区。在运算过程中，SHA-1 要用到两个缓冲区，两个缓冲区均有 5 个 32 位的寄存器。此外，运算过程中还用到 80 个 32 位字序列和一个单字的缓冲区。

③ 按 512 位的分组处理输入消息。SHA-1 运算主循环包括 4 轮，每轮 20 次操作。用到一个逻辑函数序列，每个逻辑函数输入三个 32 位字，输出一个 32 位字。

NIST 发布了三个额外的 SHA 变体，每个都有更长的消息摘要。以它们的摘要长度(以位元计算)加在原名后面来命名：SHA-256、SHA-384 和 SHA-512。它们发布于 2001 年的 FIPS PUB 180-2 草稿中，随即通过审查和评论。这些新的散列函数并没有接受像 SHA-1 一样的公众密码社群做详细的检验，所以它们的密码安全性还不被大家广泛地信任。2004 年 2 月，发布了一次 FIPS PUB 180-2 的变更通知，加入了一个额外的变种 SHA-224。

SHA-1、SHA-224、SHA-256、SHA-384 和 SHA-512 都被需要安全散列算法的美国联邦政府所应用，他们也使用其他的密码算法和协定来保护敏感的未保密资料。FIPS PUB 180-1 也鼓励私人或商业组织使用 SHA-1 加密。

2.5　密码协议

所谓协议，就是两个或者两个以上的参与者为完成某项特定的任务而采取的一系列步骤。这个定义包含如下三层含义。

① 协议自始至终是有序的过程，每一个步骤必须执行，在前一步没有执行完之前，后面的步骤不能执行。

② 协议至少需要两个参与者。

③ 通过协议必须能够完成某项任务。

密码协议(cryptographic protocol)是使用密码学完成某项特定的任务并满足安全需求的协议，又称安全协议(security protocol)。参与该协议的伙伴可能是朋友和相互信任的人，也可能是敌人和互相不信任的人。密码协议通常使用特定密码算法来实现。

常用密码协议可分为如下几种。

- 密钥建立协议：用于在两个或多个实体之间建立会话密钥。会话密钥一般是一个对称密钥，用来加密双方会话过程中的各种消息。协议可以采用对称密码体制，也可以采用非对称密码体制。可以通过一个可信的服务器为用户分发密钥，这样的密钥建立协议称为密钥分发协议；也可以通过两个用户协商，共同建立会话密钥，这样的密钥建立协议称为密钥协商协议。
- 认证协议：主要用于防止假冒攻击。包括实体认证(身份认证)协议、消息认证协议、数据源认证和数据目的认证协议等。
- 认证和密钥交换协议：将认证和密钥交换协议结合在一起，是网络通信中最普遍应用的安全协议。如 Needham-Schroeder 协议、分布认证安全服务协议和 ITU-T X.509 认证协议等。
- 数字签名协议：用于提供发送人可靠性和消息完整性的证明。

随着计算机网络应用的不断深入，还出现了保密选举协议、数字现金协议、多方计算协议等许多更深入的安全协议。

在密码协议中，经常使用对称密码、公开密钥密码、单向函数和伪随机数生成器等。

密码协议可用于保障计算机网络信息系统中秘密信息的安全传递与处理，确保网络用户能够安全、方便、透明地使用系统中的密码资源。目前，密码协议在金融系统、商务系统、政务系统、军事系统和社会生活中的应用日益普遍，而密码协议的安全性分析验证仍是一个悬而未决的问题。在实际社会中，有许多不安全的协议曾经被人们作为正确的协议长期使用，如果将其用于军事领域的密码装备中，则会直接危害到军事机密的安全性，会造成无可估量的损失。这就需要对安全协议进行充分的分析、验证，判断其是否达到预期的安全目标。

本章实验

1. 编程实现 DES 算法。
2. 编程实现 SHA-1 算法。

思考题

1. 考虑一个常用质数 $q=11$，原根 $a=2$ 的 Diffie_Hellman 方案。那么：
① 如果用户 A 的公钥为 $Y_A=9$，则 A 的私钥 X_A 是多少？
② 如果用户 B 的公钥为 $Y_B=3$，则共享的密钥 K 是多少？
2. 私钥{2,3,6,13,27,52}，取 $n=13$，$m=105$：
① 计算公钥。
② 对消息=011000　110101　101110 进行加解密。
3. 在一个使用 RSA 的公开密钥系统中，你截获了发给一个用户的密文 C=10。而且也知道 $e=5$，$n=35$，那么明文 M 是什么？
4. 在一个 RSA 系统中，一个给定用户的公钥是 $e=31$，$n=3599$，那么这个用户的私钥是什么？

第3章

数字认证技术

3.1 认证技术概述

认证(authentication),又称鉴别,是对用户身份或报文来源及内容的验证。认证包括两类：一是指在用户开始使用系统时,系统对其身份进行的确认,即对通信对象的鉴别,称为身份鉴别(认证)；二是验证网络上发送的数据(例如一个消息)的来源及其完整性,即对通信内容的鉴别,称为报文(消息)鉴别。

3.1.1 报文鉴别

报文鉴别是指在通信对等实体之间建立通信联系后,通信者对收到的信息进行验证,以保证其真实性和完整性的过程。

一般来说,这种验证过程需要确定如下内容。

- 报文是由确认的发送方产生的。
- 报文内容没有被修改过。
- 报文的接收顺序和发送顺序相同。

报文鉴别可分为报文源鉴别、报文内容鉴别和报文时间性鉴别。

(1) 报文源鉴别

报文源鉴别用于确认报文发送者的身份,可以采用多种方法实现。例如,可以通过附加在消息中的加密密文来实现报文源鉴别：发送方利用自己的私钥加密一段报文,然后发送至接收方,接收方利用对方的公钥进行解密来鉴别发送方的身份。由于系统假定只有发送方自己拥有其私钥,因此只有发送方能够产生这样的密文,接收方如果可以解密该密文即可鉴别对方身份,这就是数字签名的原理。

(2) 报文内容鉴别

内容鉴别的目的是保证通信内容没有被篡改,即保证数据的完整性,通过“消息鉴别码”

或“消息摘要”实现。鉴别的一般过程为：发送方计算出报文的“消息鉴别码”或“消息摘要”，并将其作为报头内容的一部分与报文一起传送至接收方。接收方使用同样的算法对报文内容进行计算，并将产生的鉴别码与收到的值进行比较，若相同则认为报文的内容正确、没有被篡改，否则鉴别失败。

(3) 报文时间性鉴别

报文时间性鉴别的目的是验证报文时间和顺序的正确性，即确保收到的报文和发送时的报文顺序一致，并且收到的报文不是重复的报文，可通过以下几种方法实现：

- 利用时间戳。
- 对报文进行编号。
- 使用预先给定的一次性通行字表，即每个报文使用一个预先确定且有序的通行字标识符来标识其顺序。

3.1.2 身份鉴别

身份鉴别是对终端用户的身份进行识别和验证的技术，目的是防止非法用户对计算机或网络系统的未授权访问。它是信息系统安全保密的第一道防线，也是最基本的安全措施。身份鉴别可以和报文源鉴别一同实施，如数字签名，也可分开实施，即只鉴别身份(参见 3.2 节和 3.3 节)。

一般，可以把身份鉴别方法归为以下三类。

- 密码验证：通信双方使用预先约定的通行字标识自己的身份，通过验证密码进行身份鉴别。
- 拥有物验证：通信对等实体使用拥有物，如通行证、智能卡或者密钥等进行身份鉴别。例如，数字签名中，发送者使用自己的私钥加密一段报文的消息摘要，接收者利用发送者的公钥进行解密，从而进行身份认证。
- 生物特征验证：使用消息发送者的生物特征，如指纹、声音和手写签名等进行身份鉴别。

下面详细讲述使用密码和密钥进行身份鉴别的原理和方法。

3.2 密码鉴别

3.2.1 密码与密码攻击

密码鉴别是防止未授权访问的常用方法，很多系统基于用户名和密码进行鉴别和授权，因此，密码系统本身的安全性对系统的安全性至关重要：密码一旦被破解，非法用户就可以

拥有合法用户的权限，对系统进行更改和其他操作，从而严重威胁系统安全。随着网络技术的发展以及 Internet 的广泛使用，网络上针对密码的攻击越来越频繁，密码攻击的方法也越来越多样，这使得传统的、由系统保存一个固定的用户密码并和用户输入的密码进行比较来进行身份鉴别的简单方法（称为可复用密码技术）已不能满足系统和密码安全的需要，针对密码的各种攻击使得新的密码鉴别技术不断出现。

针对密码的攻击通常有如下方法。

1. 密码猜测

攻击者进行反复尝试以猜测用户或系统的密码，如果是短密码和简单密码，这种方法是有效的。攻击者可以利用一些专门的软件强行破解用户密码，这种方法不受网段限制。

典型的密码猜测方法为字典攻击，它是一种联机攻击方法：攻击者采用字典穷举法来破解用户的密码，通常，攻击者通过一些工具程序，自动地从字典中取出一个单词，作为用户的密码，再输入给远端的主机，申请进入系统；若密码错误，就按序取出下一个单词，进行下一个尝试，并一直循环下去，直到找到正确的密码或字典的单词试完为止。由于这个破译过程由计算机程序来自动完成，因而几个小时就可以把上十万条记录的所有单词都尝试一遍。

另一种密码猜测方法为强力攻击。这是一种特殊的字典攻击，它使用字符串的全集作为字典，即穷举所有可能的密码空间。这需要很大的耐心和巨大的工作量。然而，若用户的密码较短，那么它很快就会被穷举出，因而很多系统都建议用户使用长密码。

2. 密码侦听

密码侦听即通过网络监听和数据包嗅探得到用户密码。这种方法需要在局域网内实施，具有一定的局限性，但危害性极大。监听者往往利用网络嗅探工具获取网络上的数据包，通过对数据包的分析获得用户账号和密码等信息。当前，Internet 的很多应用协议本身没有采取加密或身份认证措施，如在 Telnet、FTP、HTTP 和 SMTP 等传输协议中，用户账号和密码均以明文格式传输，此时若攻击者利用数据包截取工具便可以很容易获取用户的账号和密码。

另有一种中途截击攻击方法，它在客户端和服务器端完成“三次握手”建立连接之后，在通信过程中假冒服务器身份欺骗客户端，再假冒客户端向服务器发出恶意请求。另外，攻击者有时还会利用软件和硬件工具时刻监视系统主机的工作，等待记录用户登录信息，从而获取用户密码；或者攻击者可以编制导致缓冲区溢出错误的 SUID 程序来获得超级用户的权限。

3. 密码窃取或盗用

密码窃取或盗用一般指利用系统和网络的漏洞窃取密码。攻击者利用操作系统和网络系统的漏洞对用户密码进行窃取、盗用和分析。由于操作系统本身存在许多安全漏洞或设

计缺陷，这些漏洞和缺陷一旦被发现，黑客就可以长驱直入，获得用户密码，进而控制受害主机。例如，Unix 操作系统中，用户的基本信息存放在 passwd 文件中，用户的密码保存在 shadow 文件中，攻击者通过网络入侵或利用操作系统的漏洞即可获得该密码文件。

利用系统漏洞的典型攻击方法是特洛伊木马，这种程序可以入侵用户的计算机，它常常被伪装成工具程序或者游戏等诱使用户打开该程序，许多邮件的附件中也带有特洛伊木马程序，一旦用户打开了这些邮件的附件或者执行了这些程序之后，它们就会在计算机系统中隐藏一个可以在 Windows 系统启动时自动执行的程序。当被种植了木马的计算机连接到 Internet 上时，这个程序就会和攻击者的计算机建立连接，攻击者可以任意地修改受害计算机的参数设定、复制文件、窥视其整个硬盘中的内容，包括密码等，从而达到控制受害计算机的目的。

4. 密码分析

密码分析指对加密密码进行分析和破解。攻击者通过数据包嗅探或利用系统漏洞获得用户的密码或密码文件，很多密码是经过加密的，不能直接得到原始密码，攻击者可以利用密码分析或者采用某些针对特定加密算法的破解工具对加密密码进行分析和破解。例如，上面提到的 Unix 操作系统中用户的密码是经过 DES 算法加密后存放在 shadow 文件中的，攻击者可以使用专门破解 DES 加密法的程序来解密该密码。

5. 重放攻击

攻击者截获含有用户密码的数据包后，在不进行密码破解的情况下，直接向目标主机发送一个重复的、含有登录信息的数据包，从而登录系统。即攻击者虽然无法窃听密码，但他们可以先截取加密后的密码然后将其重放，从而利用这种方式进行有效的攻击。

为了抵抗各种针对密码的攻击，常用的方法如下。

（1）使用验证码

为了防止联机的字典或强力攻击，很多系统都采用验证码的方式，即在用户输入用户名和密码的同时，还需要输入一个额外的字符串，该字符串从服务器端产生并发送至客户端，并且在用户输入密码失败后重新生成新的值，从而可以有效防止利用密码猜测工具进行的自动登录。

（2）密码加密保护

通过采用强加密密码可以防止攻击者在有效时间内破解用户密码，这种方法不能抵御密码猜测，同时，如果没有额外措施，加密密码不能抵御重放攻击。

（3）动态密码卡

为了防止用户密码被监听和分析，目前许多应用系统采用动态密码卡进行身份鉴别。系统给用户分发一个可以有限次使用的动态密码卡，每次登录时，系统提示某种额外信息，用户根据这种提示信息在动态密码卡上找到对应的密码。每次登录系统时采用不同的密

码，密码使用一次即失效，密码全部失效后系统给用户更换新的密码卡，从而防止网络监听、密码猜测和重放攻击等。其缺点是用户密码仍在网络上传输，而且用户必须妥善保管自己的密码卡。

电子银行密码卡是动态密码卡的典型应用。电子银行密码卡以矩阵的形式存有若干字符串（即用户密码），客户在使用电子银行进行交易时，电子银行系统会随机给出一组密码卡坐标，客户根据坐标从密码中找到密码组合并输入电子银行系统进行密码鉴别。这种密码组合是动态变化的，使用者每次使用时输入的密码都不一样，交易结束后密码失效，从而在一定程度上保障电子银行交易的安全性。

（4）一次一密密码

应对密码猜测和重放攻击最有效的方法是使用一次一密密码，或称一次性密码（one time password，OTP）。在一次性密码系统中，系统利用用户输入的原始密码，结合系统生成的其他字符串，经过某种动态密码生成算法计算后，每次产生不同的密码用于系统登录，这种一次性密码难以猜测，并且用户的原始密码不在网络上传输，因此 OTP 可以抵抗字典攻击、强力攻击和重放攻击等，从而有效提高密码系统的安全性。

目前，一次一密密码技术已被广泛应用于 VPN、操作系统登录和网络访问服务器（network access server，NAS）中进行基于密码的身份鉴别。

3.2.2 验证码

为了防止针对密码的联机字典攻击和强力攻击，许多系统采用验证码和用户密码一起进行身份鉴别。不少网站为了防止用户利用密码破解程序自动注册和进行系统登录，也普遍采用了验证码技术，如图 3.1 所示。

验证码的一般工作过程和原理是客户端访问服务器时，服务器端产生一段随机码（一般是随机数或字符串，或将随机字符串经过某种算法变换或加密后形成新的随机码），然后将这段随机码以图片的形式在客户端显示，客户端输入图片中的随机码及其常规密码进行登录。服务器端需要保存这一段随机码，并通过验证用户输入的随机码和密码的正确性来进行身份鉴别。



图 3.1 验证码示例

以图片方式显示验证码的好处是可以防止客户端程序自动识别该随机码，为了达到这一目标，一般在图片中加入干扰因素以防止客户端程序进行图像识别。

验证码的实现有多种方法，对应的安全强度也不同。最简单的验证码是将服务器端生成的随机字符串保存在网页或 cookie 中传递至客户端，然后将该随机字符串在客户端以图片方式显示。基于这种方法的身份鉴别系统的安全性不高，因为攻击者可以通过截获页面

或者对 cookie 进行读取和分析来破解该验证码,因此这种方法不能很好抵御联机密码攻击。另一种简单方法是将生成的随机字符串使用服务器端程序生成一个图片,然后再将该图片传输至客户端显示。这种方式增加了程序自动分析验证码的难度,但也可以通过文本识别进行破解。

安全性高的验证码应该是用户容易识别,但机器和程序不能自动识别,并且验证码在每次用户登录时都发生变化,因此可以防御攻击者利用程序进行字典攻击和强力攻击、自动进行登录尝试等,从而有效防御密码猜测、密码分析等联机密码攻击。为了保证验证码和密码鉴别的安全性,高级的验证码不使用明文直接传递至客户端,也不直接将验证码存储在 cookie 中。

下面介绍两种高级验证码的实现方法:一是加密后的验证码在 cookie 中传递至客户端,二是验证码不进行传递,而是保存在服务器端。

1. 验证码在 cookie 中

这种方式中,验证码的实现过程描述如下。

① 服务器生成一个随机字符串。

② 服务器端使用某种加密算法 f(该算法不可逆,破解难度高)将该字符串转化成为验证码,再转化成图片。

③ 图片被发送到客户端,并将图片显示在客户端登录界面上。

④ 客户端输入用户名和原始密码,同时,按照图片内容输入验证码,进行登录。

⑤ 服务器检查验证码和用户密码是否正确。

2. 验证码保存在服务器端

这种方式中,验证码的实现过程描述如下。

① 服务器根据用户相关信息(如用户 IP、用户身份标识 SID 等)生成一个随机字符串。

② 服务器端使用某种加密算法 f 将字符串转化成为验证码。

③ 验证码被保存在服务器的本地数据库中(通常该数据库包括 session,有关用户 IP 等信息),并由一个序列号 seq 指向该用户对应的验证码。

④ 这个序列号 seq 被作为 cookie 发送至客户端,并以图片方式显示。

⑤ 客户端输入验证码。

⑥ 服务器按照客户端输入的序列号读取数据库中所期望中的验证码。服务器还可以对 seq 与 session ID 之间的关系进行验证。

⑦ 用户进行了验证操作或重新获取验证码后,服务器将原来数据库中的验证码用新的值替换。

验证码可以在一定程度上增加攻击者使用自动程序进行密码攻击的难度,但从理论上讲,采用验证码方式的鉴别系统不能完全抵御字典攻击和强力攻击。同时,利用验证码和普

通密码系统进行身份鉴别时,用户密码仍在网络上传输,并且密码是重复用的,无法抵御密码监听、密码分析等密码攻击。

3.2.3　一次一密密码

一次一密密码协议 OTP 由 RFC 1938 定义。

一次一密密码系统由终端用户、远程服务器和密码计算器构成。首先,在用户和远程服务器之间建立一个共享通行字——相当于传统密码系统中的"密码",在一次性密码系统中该通行字称为"通行短语"。同时,它们之间还应具备一种相同的"密码计算器",该密码计算器实际上是某种动态密码算法的硬件或软件实现,它的作用是生成一次性密码。

当用户向服务器发出连接请求时,服务器向用户提出挑战(challenge)。挑战由两部分组成:一部分是种子值(seed),它是分配给用户的在系统内具有唯一性的一个数值,也就是说,一个种子对应于一个用户,同时它是非保密的;另一部分是迭代值(iteration),它是服务器临时产生的一个数值,与通行短语和种子值不同的是,迭代值总是不断变化的。

用户收到挑战后,将种子值、迭代值和通行短语输入到自己的"密码计算器"中进行计算,并把结果作为应答返回服务器。服务器暂存从用户那里收到的应答,由于服务器知道用户的通行短语、种子值和迭代值,因此它能够计算出和用户相同的、正确的应答值,通过比较这两个值,服务器就可以鉴别用户的身份。

OTP 的验证过程如图 3.2 所示。可以看出,在一次一密密码系统中,用户通过网络传递至服务器的密码是由种子值、迭代值和通行短语在密码计算器作用下的计算结果,用户本身的通行短语没有在网络上传输。只要密码计算器中的算法足够复杂,就很难从中破解出原始的通行短语,从而有效地抵御网络窃听攻击。又因为迭代值总是不断变化的,比如当身份认证成功时,将用户的迭代值自动减 1,这使得下一次用户登录时使用的密码鉴别信息和前一次不同(一次性密码技术由此得名),从而可以有效抵御重放攻击。总之,与可重用密码技术的单因子(用户密码)鉴别不同,一次性密码技术是一种多因子(包括种子值、迭代值和通行短语)鉴别技术,其中引入的不确定因子使得它比传统密码系统更为安全。

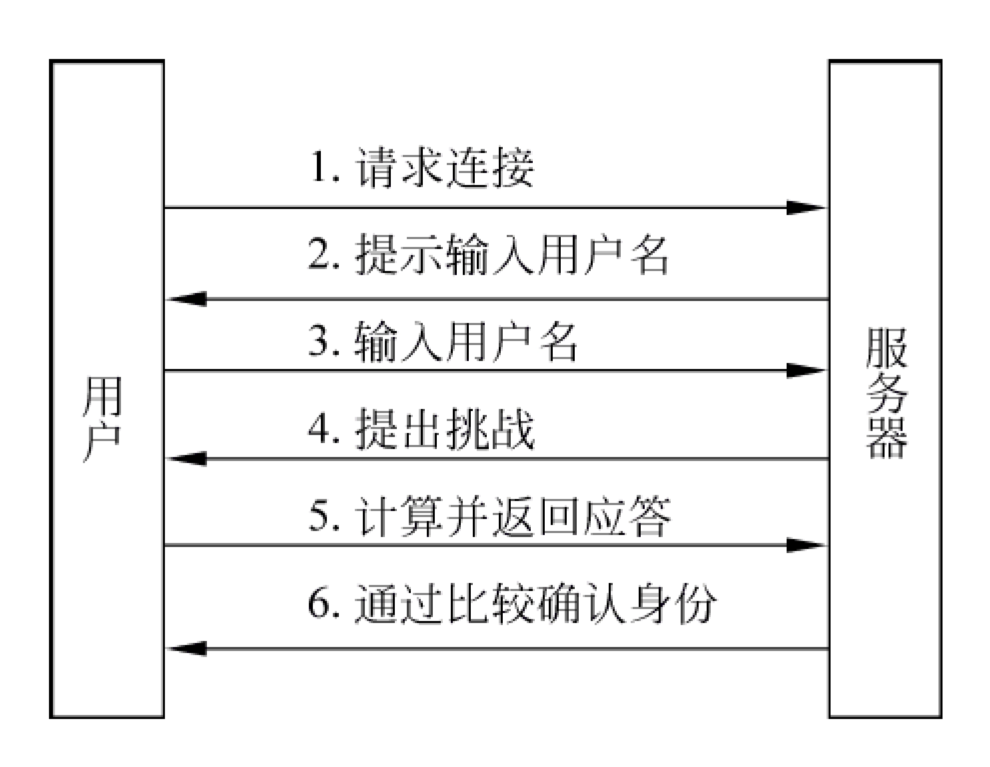


图 3.2　一次一密密码验证过程

OTP 技术的具体实现包括目前广泛使用的、由 Bellcore 研制的 S/KEY 系统,以及在其基础上开发的 IPIE(ine-time passwords in everything)系统。OTP 可广泛应用于 VPN、操作系统登录和远程访问服务器等。

在实施时,OTP 系统可以和应用系统集成,即把 OTP 和客户端及服务器端软件集成,

密码的生成和鉴别过程对用户透明：用户只需输入原始密码（PIN 密码），一次性密码的计算和输入由客户端系统自动完成；也可以将 OTP 和应用系统分离，即采用专门的 OTP 计算器产生一次性密码，用户登录系统时输入原始密码的同时，需要输入 OTP 产生的一次性密码（某些系统中称为 token）。后一种情况，OTP 计算器可以内嵌在一个硬件设备中分发给用户。本章 3.5 节将介绍 S/KEY 系统中一次性密码的实现和应用。

3.2.4 基于挑战/应答的鉴别

基于挑战/应答（challenge/response）方式的身份认证机制属于密码鉴别的一种，其特点是密码不在网络上传输。该认证机制中，每次认证时认证服务器（认证者）向客户端（被认证者）发送一个不同的“挑战”字串，客户端程序收到这个“挑战”字串后，按照双方事先协商好的方式做出相应的“应答”。“挑战”字符串相当于认证方对被认证方的质询，而应答是针对该质询的回答。认证方通过检查该应答是否正确来确认对方的身份。

如图 3.3 所示，一个典型的基于挑战/应答身份的认证过程描述如下：

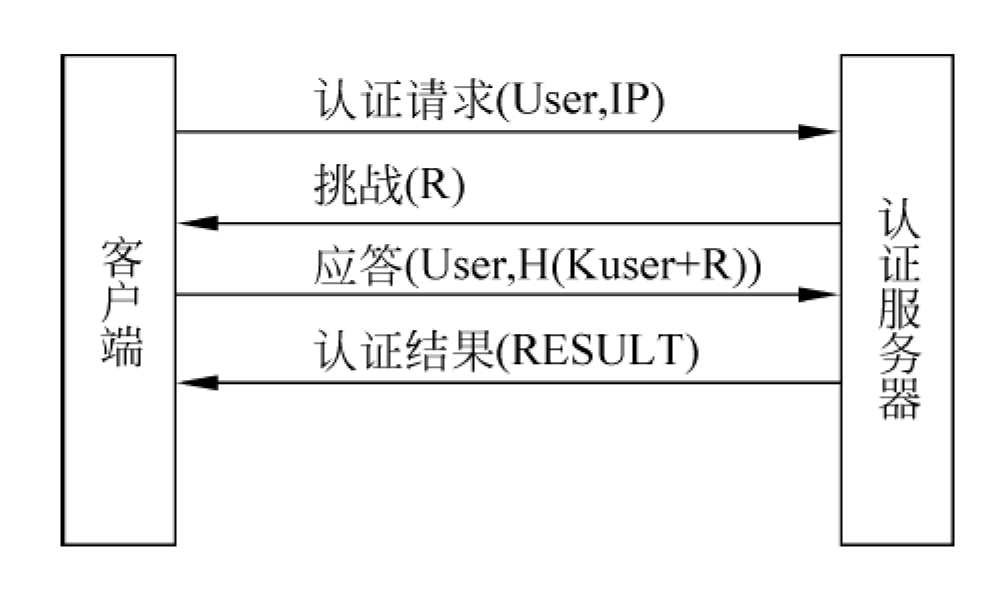


图 3.3 基于挑战/应答的鉴别过程

① 客户端向认证服务器发出请求，要求进行身份认证。

② 认证服务器从用户数据库中查询该用户是否为合法用户，若不是，则不做进一步处理。

③ 认证服务器内部产生一个随机数，即挑战字符串。作为对客户端的“质询”或“提问”，发送至客户端。

④ 客户端将用户名字和随机数合并，将双方共享的密码或密钥作为输入，使用单向 Hash 函数（例如 MD5 算法）生成一个字符串作为应答。

⑤ 认证服务器将应答串与自己的计算结果比较，若两者相同，则通过认证，否则认证失败。

⑥ 认证服务器通知客户认证成功或失败。

之后，认证请求由客户不定时地发起，两次认证的时间间隔应适当选取，太短会给网络、客户端和认证服务器带来过多的资源耗费，时间间隔太长则不能抵御 IP 地址盗用等网络攻击，一般可为 1～2 分钟。

1. 密钥的分配和管理

密钥的分配由维护模块负责，当用户进行注册时，自行设定自己的密码字。用户的密钥由密码字生成。

一个密码字必须经过两次密码字检查。第一次由注册程序检查，强制密码字必须有足

够的长度(如 8 个字符)。密码字被加密后送入数据库中,这个密码字标记为“未检查的”。第二次,由离线的密码字检查工具进行检查,将弱密码字进行标记,当下一次用户认证时,认证服务器将强制用户修改密码字。

密钥的在线修改由认证服务器完成,它的过程与认证过程基本类似。

2. 安全性分析

(1) 网络侦听(sniffer)

认证过程中,密钥和密码字不在网络上传输,所以网络侦听攻击无效。而在密钥的在线修改过程中,新密码字使用旧的密钥加密传送,网络侦听攻击仍然无效。

由于采用了单向 Hash 函数来对密码字和随机数进行处理,侦听者很难从侦听到的报文得到用户的密码字。

(2) “重放”攻击(replay)

由于认证服务器每次选取的随机数不同,即 Challenge 值不同,所以无法通过“重放”前面侦听到的认证报文来完成以后的认证。

(3) 密码字猜测(password guessing)

侦听者在获知认证算法后,可以对用户的密码字进行猜测:使用计算机猜测密码字,利用得到的报文进行验证。这种攻击办法直接有效,特别是当用户的密码字有缺陷时,比如密码字短、使用名字做密码字、使用一个单词做密码字(可以使用字典攻击)等。

对付这种攻击的办法是使用一个很长的密码字,并避免使用用户名字中的字,避免使用一个单词做密码字等。系统本身对密码字要求严格,首先密码字必须取得足够长。用户的登记和修改密码字的程序强制用户的密码字长度。其次,可以利用离线的密码字检查工具,将弱密码字标记,强制用户限期修改。这样,用户的密码字就有足够的抗攻击强度。

(4) 跟踪地址攻击

即攻击者在看到用户认证后,设法将自己的机器地址更改为用户的 IP 地址,从而冒充用户上网。但由于攻击者无法完成后续的认证,在 1~2 分钟内,攻击失效。

以上描述的是基于挑战/应答身份认证机制一个示例过程,不同的协议在具体实现上有所不同,如产生“挑战”和应答的算法以及认证的频率等由具体协议确定。质询握手认证协议(challenge handshake authentication protocol,CHAP)是基于挑战/应答身份认证的一种具体协议(见 3.5 节)。

基于挑战/应答的鉴别和 OTP 的思想类似,两者均不在网络上传输密码,而是发送经过加密的密文,该密文和密码以及认证者原始发送的一个字符串相关。区别是两者防止密码攻击的方法不同:基于挑战/应答的鉴别每次鉴别时通过改变“挑战字”的值来抵御“重放”攻击,而 OTP 则是改变密文的内容(每次使用不同的密文)来增加密码系统的安全性。

3.3 基于密钥的鉴别

3.3.1 基于对称密钥的鉴别

对称密钥加密体制中,通信双方共享一对密钥。假设只有通信双方知道该密钥,则该对称密钥可以作为用户身份鉴别的依据。最简单的基于对称密钥的鉴别过程如图 3.4 所示。

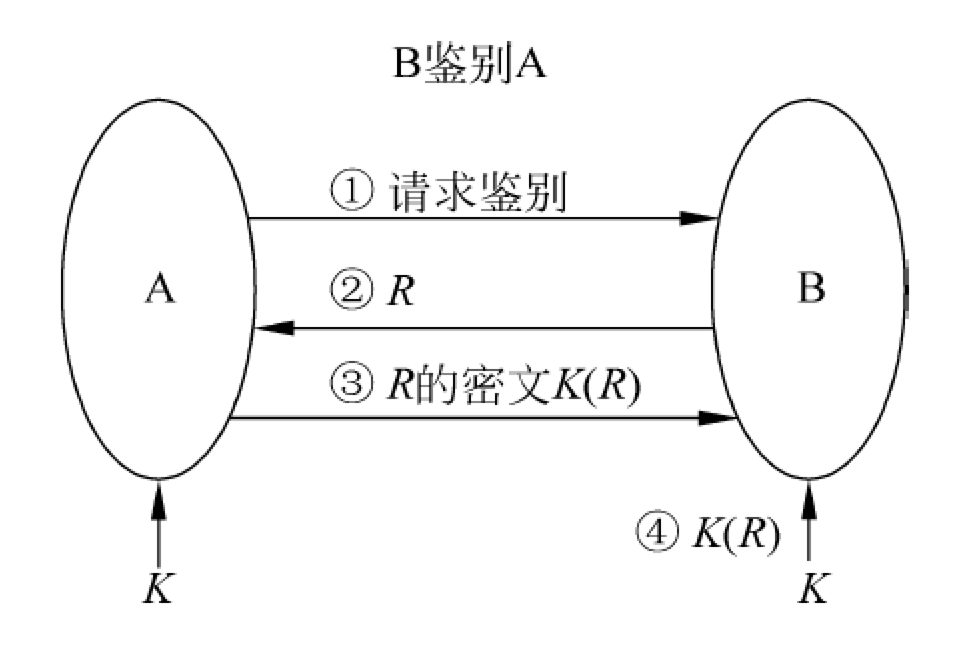


图 3.4 基于对称密钥的单向鉴别

假设通信双方 A 和 B 拥有事先协商产生的一对对称密钥 K,B 欲对 A 的身份进行鉴别,鉴别过程描述如下。

① A 向 B 发送一条消息,要求对方鉴别自己的身份。

② B 向 A 发送一个随机数 R。

③ A 使用对称密钥 K 将该随机数 R 加密后形成密文 $K(R)$,然后将 $K(R)$传输给 B。

④ B 使用对称密钥 K 解密该密文,如果能够还原出 R,则成功鉴别 A 的身份。

上述利用对称密钥进行身份鉴别的依据是:由于假定只有 A 拥有自己的私钥 K,因此也只有 A 能够生成密文 $K(R)$,如果 B 通过解密该密文得到 R,即可鉴别 A 的身份。

上述鉴别过程可用于双向鉴别,即 B 鉴别 A 身份的同时,A 对 B 的身份进行鉴别。鉴别过程如图 3.5 所示。

① A 向 B 发送一个随机数 R_A。

② B 向 A 发送另一个随机数 R_B,同时将 R_A 使用对称密钥 K 加密后形成密文 $K(R_A)$发送至 A。

③ A 使用对称密钥解密该密文,如果能够还原出 R_A,则成功鉴别 B 的身份。

④ A 将②中收到的随机数 R_B 使用对称密钥 K 加密后形成密文 $K(R_B)$发送至 B。

⑤ B 解密该密文,如果能够还原出 R_B,则成功鉴别 A 的身份。

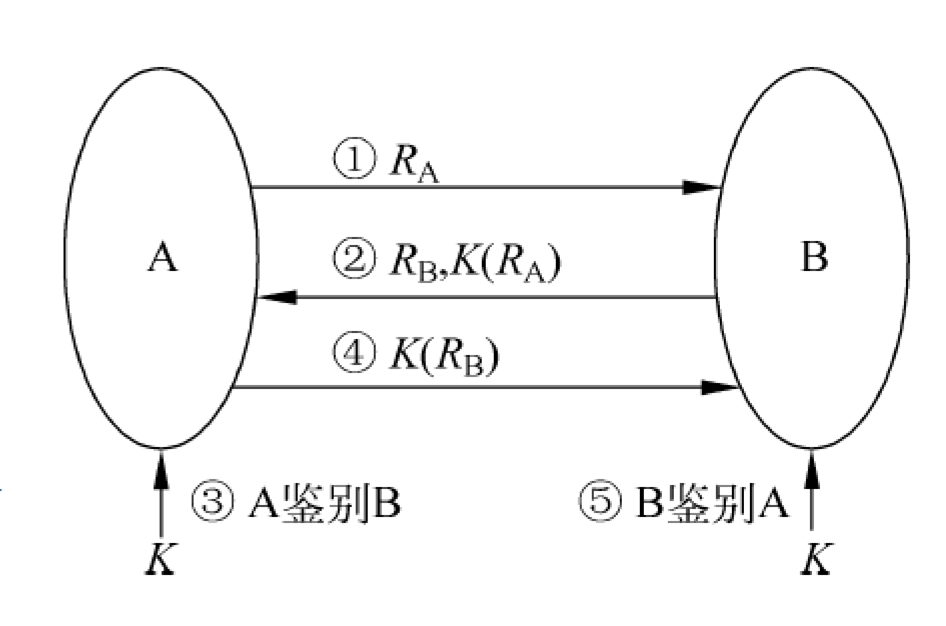


图 3.5 基于对称密钥的双向鉴别

上述鉴别过程中,随机数 R 采用明文在网络上传输,如果第三方截获随机数 R 和其加密后的密文 $K(R)$,则可以对该鉴别系统进行密码分析。为了避免这种密码分析,可以对上述鉴别系统进行简单改进,如图 3.6 所示,在第 2 步中将随机数 R 使用对称密钥加密后传输至 A;在第三步中,为了防止“重放”攻击,可以给密文加上时间戳,或者加入某些附加信

息。双向鉴别过程可采用同样的方法,以提高系统的安全性。

3.3.2 基于非对称密钥的鉴别

利用非对称密码体系中的私钥也可以进行身份鉴别。例如,用户 B 对用户 A 进行单向鉴别之前,A 将自己的公钥传递给 B,鉴别过程如图 3.7 所示。

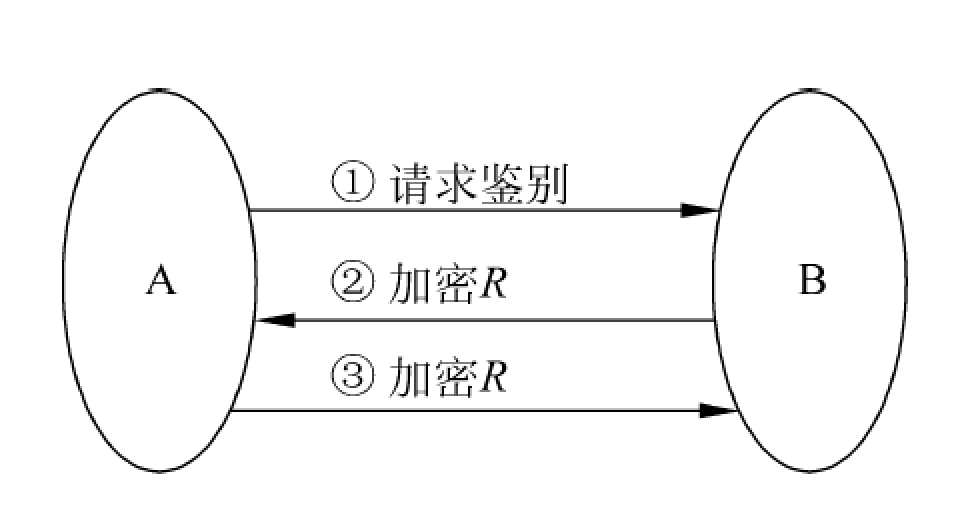


图 3.6 改进的对称密钥单向鉴别

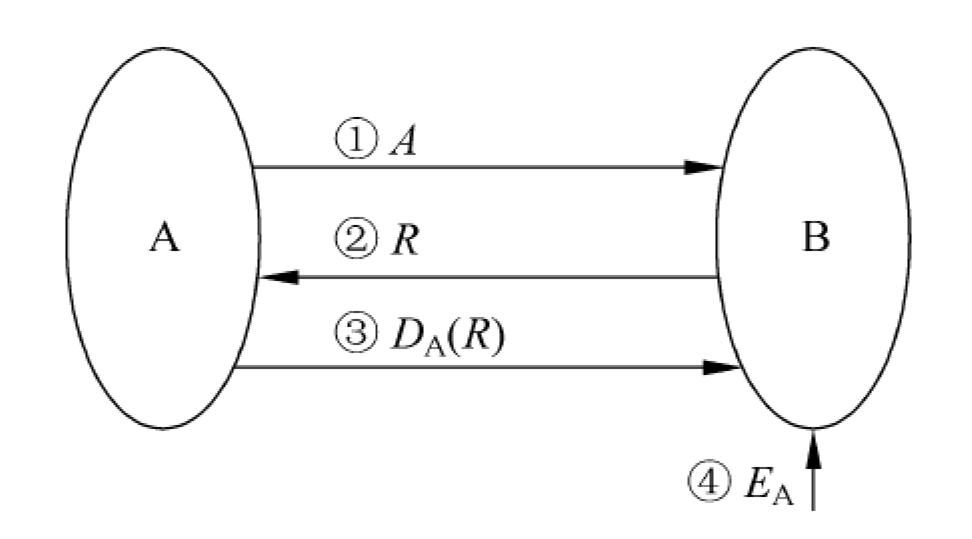


图 3.7 基于公钥的单向鉴别机制

① A 向 B 发送一条消息,要求进行身份鉴别。

② B 向 A 发送一个随机数 R。

③ A 用自己的私钥 D_A 加密该随机数得到密文 $D_A(R)$,并将该密文传输至 B。

④ B 使用 A 的公钥 E_A 解密该密文,如果能够还原出 R,则成功鉴别 A 的身份。

上述利用非对称密钥进行身份鉴别的依据是由于假定只有 A 拥有自己的私钥 E_A,因此也只有 A 能够生成密文 $E_A(R)$,如果 B 通过解密该密文得到 R,即可鉴别 A 的身份。即利用"私钥加密的唯一性"进行身份鉴别。和利用对称密钥进行身份鉴别相同,上述鉴别过程可用于双向鉴别。

此外,利用"私钥解密的唯一性"也可进行身份鉴别,如图 3.8 所示,利用私钥解密进行双向鉴别的例子可描述如下。

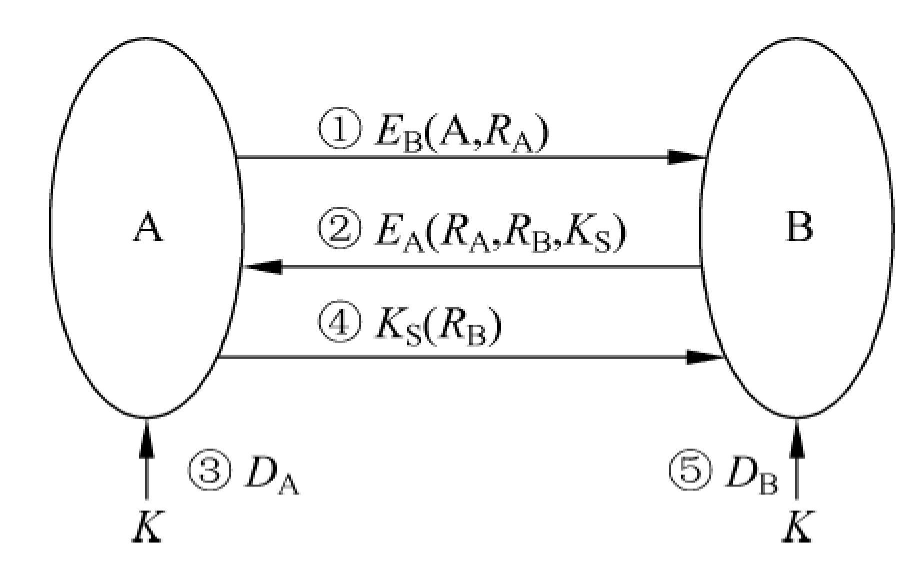


图 3.8 基于公钥的双向鉴别

① A 产生一个随机数 R_A,将其和另一个变量 A 一起使用 B 的公钥进行加密,形成密文 $E_B(A,R_A)$,将此密文传输至 B。

② B 利用自己的私钥 D_B 解密该密文后还原变量 R_A,并将其和另一个随机数 R_B,一个双方的共享密钥 K_S 一起,使用 A 的公钥 E_A 加密后形成密文 $E_A(R_A,R_B,K_S)$,将该密文传输至 A。

③ A 利用自己的私钥 D_A 解密该密文,如果能还原出 R_A,则其成功鉴别 B 的身份。

④ A 将解密的随机数 R_B 使用对称密钥 K_S 加密后传输至 B。

⑤ B 利用对称密钥 K_S 解密该密文后,如果能还原出 R_B,则成功鉴别 A 的身份。

在②中,由于只有 B 具有自己的私钥 D_B,因此只有 B 能够解密密文 $E_B(A,R_A)$,还原出

R_A,因此在③中,A 根据这种私钥解密的唯一性可以确定 B 的身份。同理,在④中,只有 A 能够利用自己的私钥 D_A 解密密文 $E_A(R_A, R_B, K_S)$,还原出 R_B,因此,B 可以根据这种私钥解密的唯一性对 A 进行身份鉴别。上述鉴别过程同时产生一个共享密钥 K_S,它可以在双方进行身份鉴别之后进行数据加密,该对称密钥也称为会话密钥。

3.3.3 基于第三方的鉴别

基于第三方的鉴别是指通信双发不直接进行身份鉴别,而是利用一个双方都信任的第三方来鉴别身份,例如使用密钥分发中心(key distribution center,KDC)进行身份鉴别。

如前所述,在使用加密机制的网络中,节点之间需要使用对方的密钥(或双方共享的密钥)进行身份鉴别,因此,为了和多个实体之间进行身份鉴别,每个节点需要保存多个对称或非对称密钥。为了减少节点保存密钥的数量,可以采用一个可信的第三方——密钥分发中心 KDC——进行密钥的保存、分配和身份鉴别,该方法基于对称加密。

基于 KDC 的鉴别中,所有合法用户的私钥都保存在 KDC 处。设两个用户 A、B 需要相互鉴别,用户 A 和 KDC 之间进行通信的对称密钥为 K_A,用户 B 和 KDC 之间进行通信的对称密钥为 K_B。A 和 B 之间加密通信的会话密钥为 K_S,此密钥是在鉴别发起方 A 向 KDC 申请要求与 B 进行身份鉴别时,由 KDC 临时产生的。如图 3.9 所示,基于 KDC 鉴别过程的例子可描述如下。

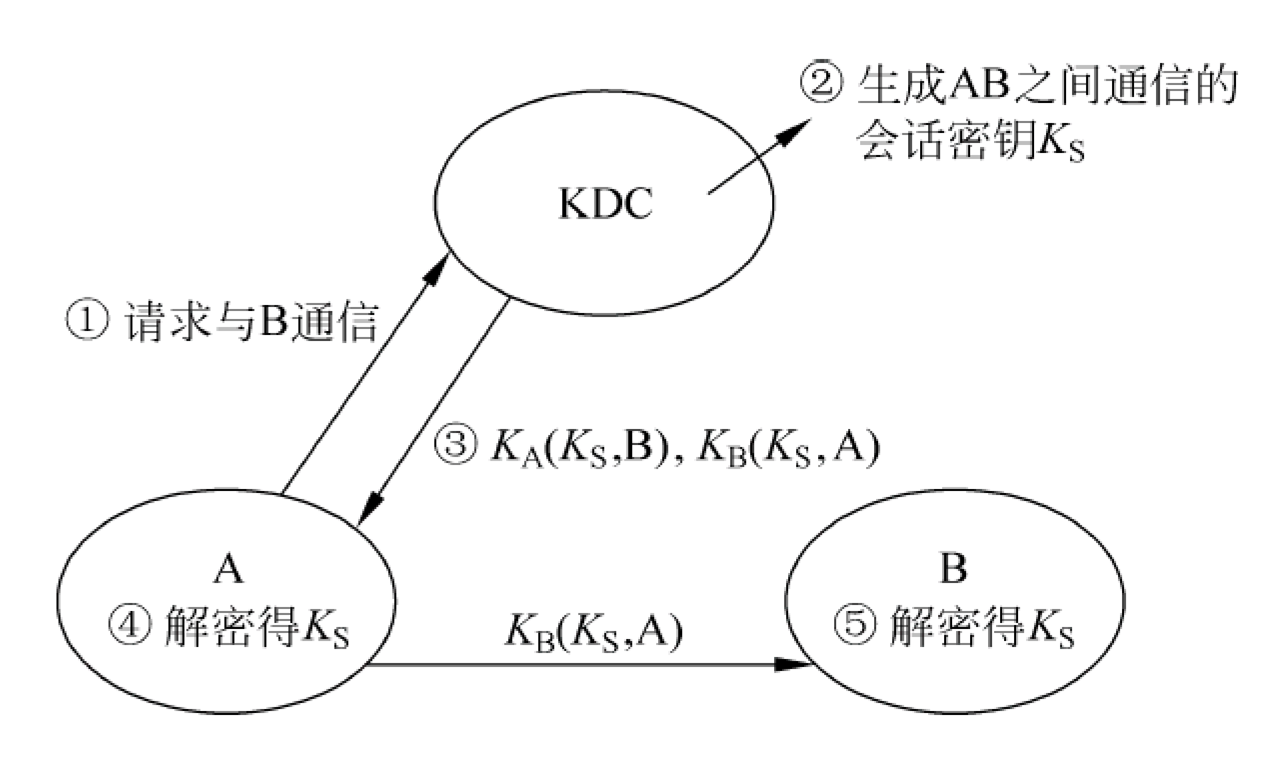


图 3.9 基于 KDC 的鉴别过程

① A 向 KDC 发送一条消息,请求与 B 进行身份鉴别和加密通信。

② KDC 为 A、B 之间加密通信生成一对对称密钥 K_S,生成和用户 A 和 KDC 之间进行通信的对称密钥 K_A,用户 B 和 KDC 之间进行通信的对称密钥 K_B。

③ KDC 使用 K_A 加密 K_S 和代表 B 身份的变量,形成密文 $K_A(K_S, B)$;同时,使用 K_B 加密 K_S 和代表 A 身份的变量形成密文 $K_B(K_S, A)$,然后将此两密文发送至 A。

④ A 利用 K_A 解密 $K_A(K_S, B)$,取得 K_S,和 KDC 之间进行了身份鉴别。

⑤ B 利用 K_B 解密 $K_B(K_S, A)$,取得 K_S,和 KDC 之间进行了身份鉴别。

随后，A、B 之间利用对称密钥 K_S 进行加密通信。

在④中，由于只有 A 拥有自己的私钥 K_A，因此只有 A 能够解密密文 $K_A(K_S, B)$得到 K_S；同理，在⑤中，只有 B 拥有自己的私钥 K_B，因此只有 B 能够解密密文 $K_B(K_S, A)$得到 K_S。因此，通过私钥解密的唯一性，KDC 信任 A 和 B，又由于 A、B 均信任 KDC，因此，A、B 之间可以互相鉴别对方的身份。

基于 KDC 的鉴别算法包括 Needham-Schroeder 方法以及扩展的 Needham-Schroeder 方法。

3.4 数字签名

数字签名提供一种安全机制，对网络上传输的消息提供报文源鉴别和消息完整性服务。一个完整的数字签名应提供如下安全能力。

- 报文源鉴别：接收方可以鉴别发送方的身份。
- 不可否认性：发送方不能否认曾发送过该报文。
- 数据完整性：接收方可以确认收到的消息是未经篡改的。
- 接收方自己不能伪造该报文。

传统文字签名的特点是签名和被签名的文件在物理上不可分割、签名者不能否认自己的签名、签名不能被伪造并且容易被验证等。从功能上讲，可以认为数字签名是传统签名的数字化，数字签名可以和被签文件"绑定"，证明文件的确来自签名者，签名者不能否认自己的签名，签名容易被验证并且签名不能被伪造等。

从本质上讲，数字签名是一种附加在原文件之上的附加电子信息，通常是能够确认报文来源即签名者身份的信息。数字签名可以通过对称密码体制实现，也可以通过非对称密码体制实施。由于非对称密码体制具有公钥可以公开并被分发的特性，使得基于公钥的数字签名被广泛使用。

基于公钥的数字签名使用公钥密码体制对文件进行签名，如图 3.10 所示，最简单的公钥签名过程可描述如下。

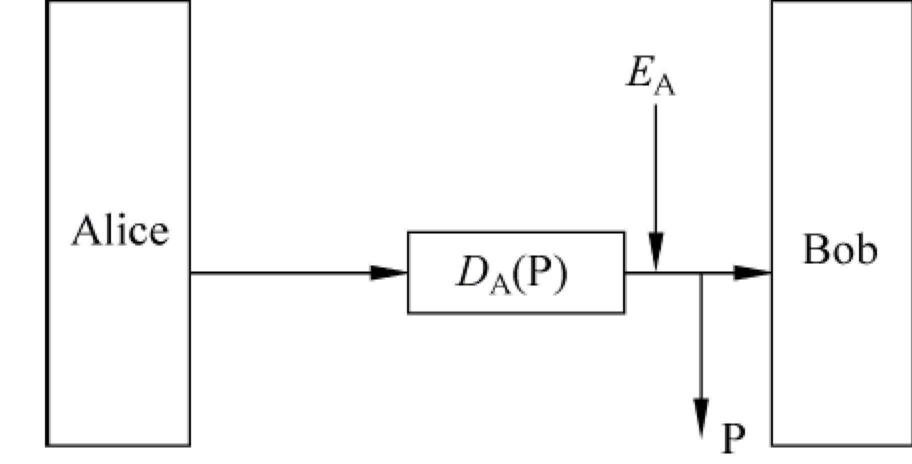


图 3.10 简单的基于公钥的签名

① Alice 用她的私钥 D_A 对文件 P 加密后形成密文 D_A(P)，由于只有 Alice 拥有自己的私钥，因此该签名可以认为是证明 Alice 身份的电子信息。这一加密过程也称为 Alice 对原始文件进行签名。

② Alice 将签名的文件传输至 Bob。

③ Bob 用 Alice 的公钥 E_A 解密该密文，从而验证签名，也就完成了对报文来源的鉴别(证明报文的确来自 Alice)。

为防止签名被重放，文件签名应该具有时间标记，因为如果数字签名不包括时间标记，那么签名文件就有可能被重复利用。

利用公钥签名的算法应满足条件：$E(D(\mathrm{P}))=\mathrm{P}$。即利用私钥加密后的密文可以使用公钥解密。

上述利用公钥进行签名的特点是签名需要对整个报文进行非对称加密，加解密效率低。

为了提高签名系统的效率，目前普遍采用将加密和签名分开的方法，如图 3.11 所示，其工作过程描述如下。

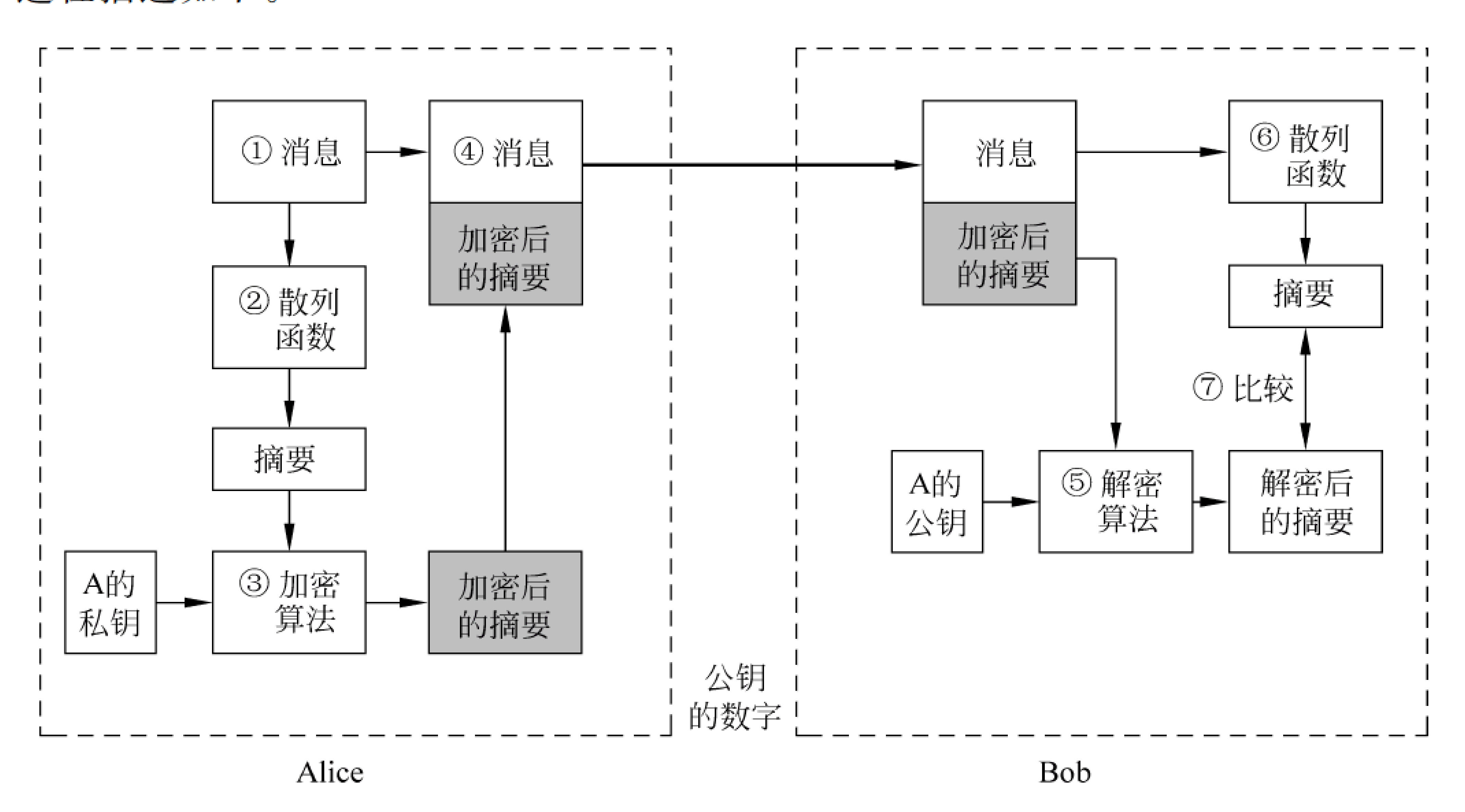


图 3.11　基于公钥的数字签名过程

① Alice 准备好要发送的明文消息。

② Alice 对该消息进行哈希(Hash)运算，得到一个消息摘要。

③ Alice 用自己的私钥对消息摘要进行加密得到 Alice 的数字签名。

④ Alice 将签名和原始明文消息一起传送给 Bob。

⑤ Bob 使用 Alice 的公钥对 Alice 的数字签名进行解密，验证 Alice 的签名，同时得到消息摘要。

⑥ Bob 用相同的 Hash 算法对收到的明文再进行一次 Hash 运算，得到一个消息摘要。

⑦ Bob 将⑤中得到的消息摘要和⑥中新产生的消息摘要进行比较，如果一致，说明收到的信息没有被修改过。

上述签名过程提供了消息完整性和报文源鉴别服务。其中报文源鉴别通过⑤中 Bob 对 Alice 的签名进行验证获得，而消息完整性验证通过⑦中对两次消息摘要进行比较来保证。

上述签名过程仅对一个固定长度的消息摘要采用私钥(公钥密码体制)加密，而不对整

个报文进行加密,因此提高了签名系统的效率。但签名过程不能保证消息的机密性。实际应用中,如果需要保证消息的机密性、真实性(报文源鉴别)、不可否认性和数据完整性,可以采用公钥签名和对称密钥加密相结合的方式。工作过程描述如下。

前三步和上述签名过程相同。

④ Alice 随机产生一个对称加密密钥(例如可以使用 3DES 密钥),并用此密钥对要发送的原始明文信息进行加密,形成密文。

⑤ Alice 用 Bob 的公钥对刚才随机产生的加密密钥进行加密,将加密后的对称密钥连同密文以及③中 Alice 对消息摘要的签名信息一起发送给 Bob。

⑥ Bob 收到 Alice 发送的密文和加过密的对称密钥,使用自己的私钥对加密的对称密钥进行解密,得到该对称密钥。

⑦ Bob 使用该对称密钥对收到的密文进行解密,得到原始明文信息。

⑧ Bob 使用 Alice 的公钥对 Alice 的数字签名进行解密,得到消息摘要。

⑨ Bob 使用相同的 Hash 算法对收到的明文再进行一次 Hash 运算,得到一个新的消息摘要。

⑩ Bob 将收到的消息摘要和新产生的消息摘要进行比较,如果一致,说明收到的信息没有被修改过。

基于公钥的数字签名使用的常用密码算法包括:

- Diffie-Hellman。
- RSA。
- 椭圆曲线密码体制(ellipse curve cryptosystem,ECC)
- DSS。

3.5 认证技术的应用

3.5.1 PPP 中的认证

PPP 协议(point to point protocol)是 Internet 工程任务组(internet engineering task force,IETF)推出的点到点类型线路的数据链路层协议。它解决了 SLIP 中的问题,并成为正式的因特网标准。

PPP 支持在各种物理类型的点到点串行线路上传输上层协议报文。PPP 有很多丰富的可选特性,如支持多协议,提供可选的身份认证服务,可以以各种方式压缩数据,支持动态地址协商,支持多链路捆绑等。这些丰富的选项增强了 PPP 的功能。同时,不论是异步拨号线路还是路由器之间的同步链路均可使用。因此,应用十分广泛。

为了在点到点链路上建立通信,PPP 链路的每一端在链路建立阶段必须首先发送链路

控制协议(link control protocol,LCP)包进行数据链路配置。链路建立之后,PPP 提供可选的认证阶段,可以在进入(network layer protocol,NLP)阶段之前实施认证。

在默认情况下,认证不是必须的,如果需要链路认证,PPP 必须在链路建立阶段指定"认证协议配置"选项。这些认证协议主要用于主机和路由器,这些主机和路由器一般通过交换电路线或者拨号线连在 PPP 网络服务器上,但是也可以通过专线实现。服务器可以用主机或路由器的连接身份作为网络层协商的选项。

PPP 提供了两种可选的身份认证方法,包括:

- 密码认证协议(password authentication protocol,PAP)。
- 质询握手认证协议(challenge handshake authentication protocol,CHAP)。
- 扩展认证协议(extensible authentication protocol,EAP)。

1. PAP 认证

PAP 是一个简单的、实用的身份认证协议。PAP 认证过程以及 PPP PAP 的工作过程分别如图 3.12 和图 3.13 所示。采用 PPP 协议的对等实体首先使用 LCP 协议确定双方的认证方式,协商使用 PAP 进行身份认证。远程访问服务器(认证者)的数据库中保存客户端(被认证者)的用户名和密码,客户端输入自己的用户名和密码后,服务器端在其数据库中进行比对,根据比对结果确定是否通过验证。

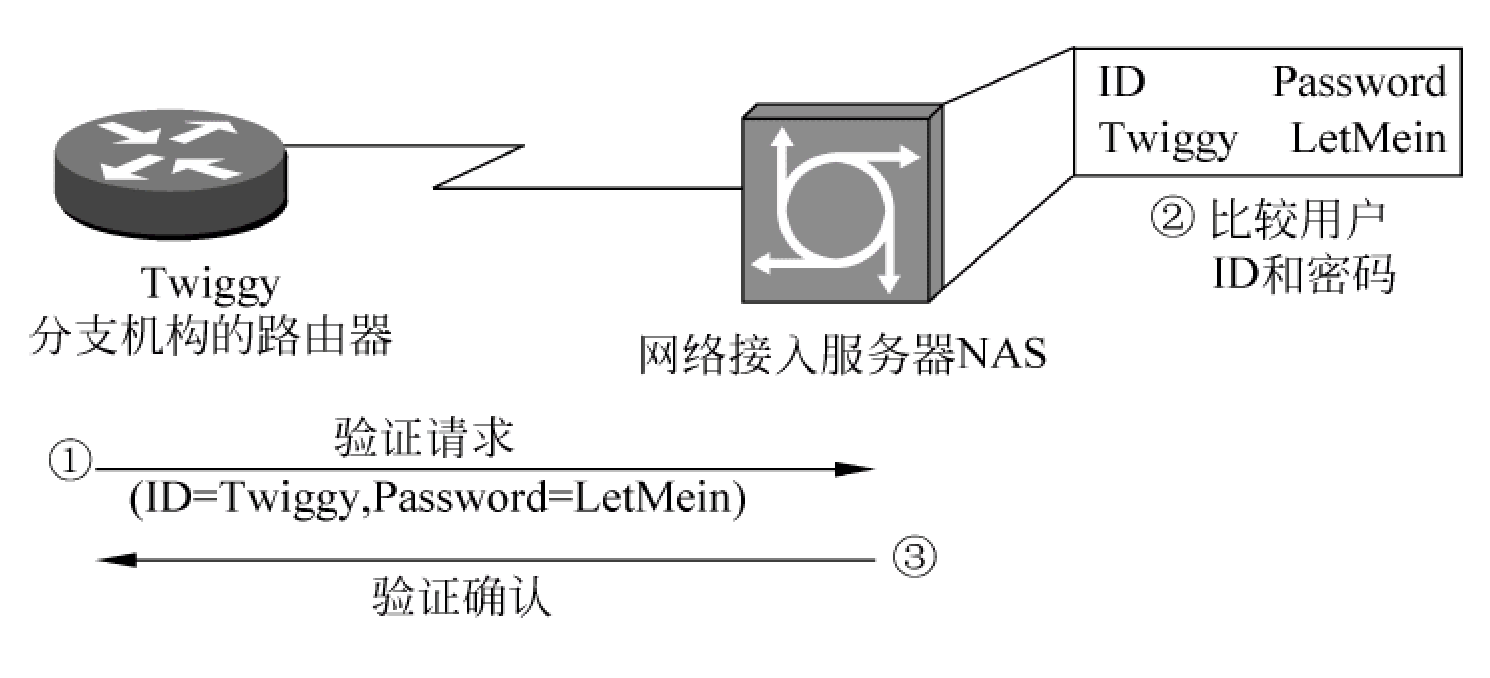


图 3.12 PAP 认证过程

PAP 认证进程只在双方的通信链路建立初期进行。如果认证成功,在通信过程中不再进行认证。如果认证失败,则直接释放链路。

PAP 的弱点是用户名和密码是明文发送的,有可能被协议分析软件捕获而导致安全问题。但是,因为认证只在链路建立初期进行,节省了宝贵的链路带宽。现在的许多拨号网络采用 PAP 协议进行身份认证,并且系统的用户名和密码是公开的,服务器端只根据链路建立的时间收费,收费是针对客户端的电话号码进行的,攻击者截获密码已经没有实际意义,因此使用简单的验证机制是适用的。

PAP 认证可以在一方进行,即由一方认证另一方身份,也可以进行双向身份认证。这时,要求被认证的双方都要通过对方的认证程序。否则,无法建立两者之间的链路。



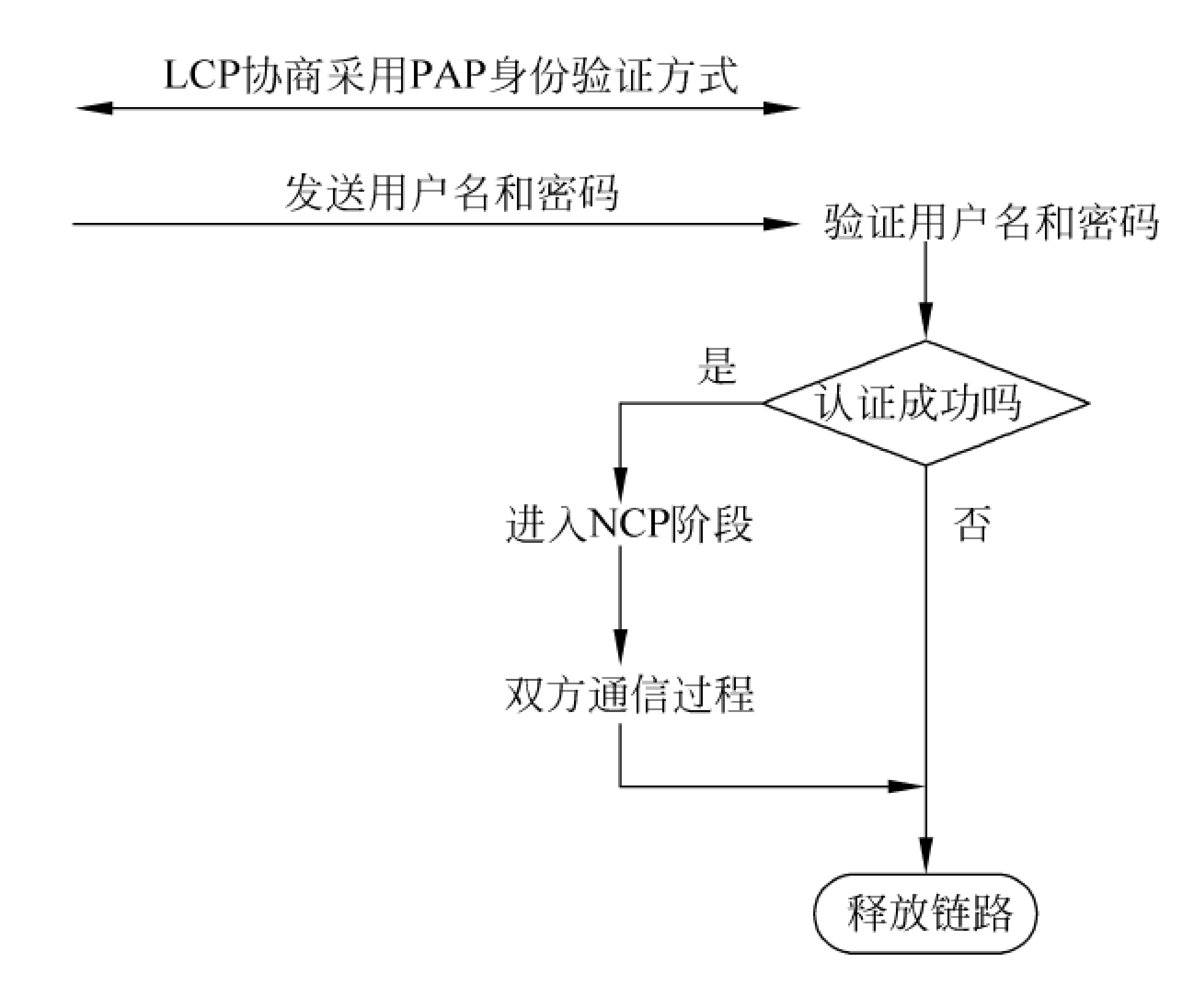


图 3.13 PPP PAP 的工作过程

下面讲述一个实际的使用 PAP 进行 PPP 认证的例子，以单方认证为例分析 PAP 协议的认证及配置过程。

如图 3.14 所示，在两个路由器之间进行 PAP 认证。两个路由器 Router A 和 Router B 双方均封装了 PPP 协议且要求进行 PAP 身份认证，同时它们之间的链路在物理层已激活，此时认证服务器会不停地发送身份认证要求，直到身份认证成功。

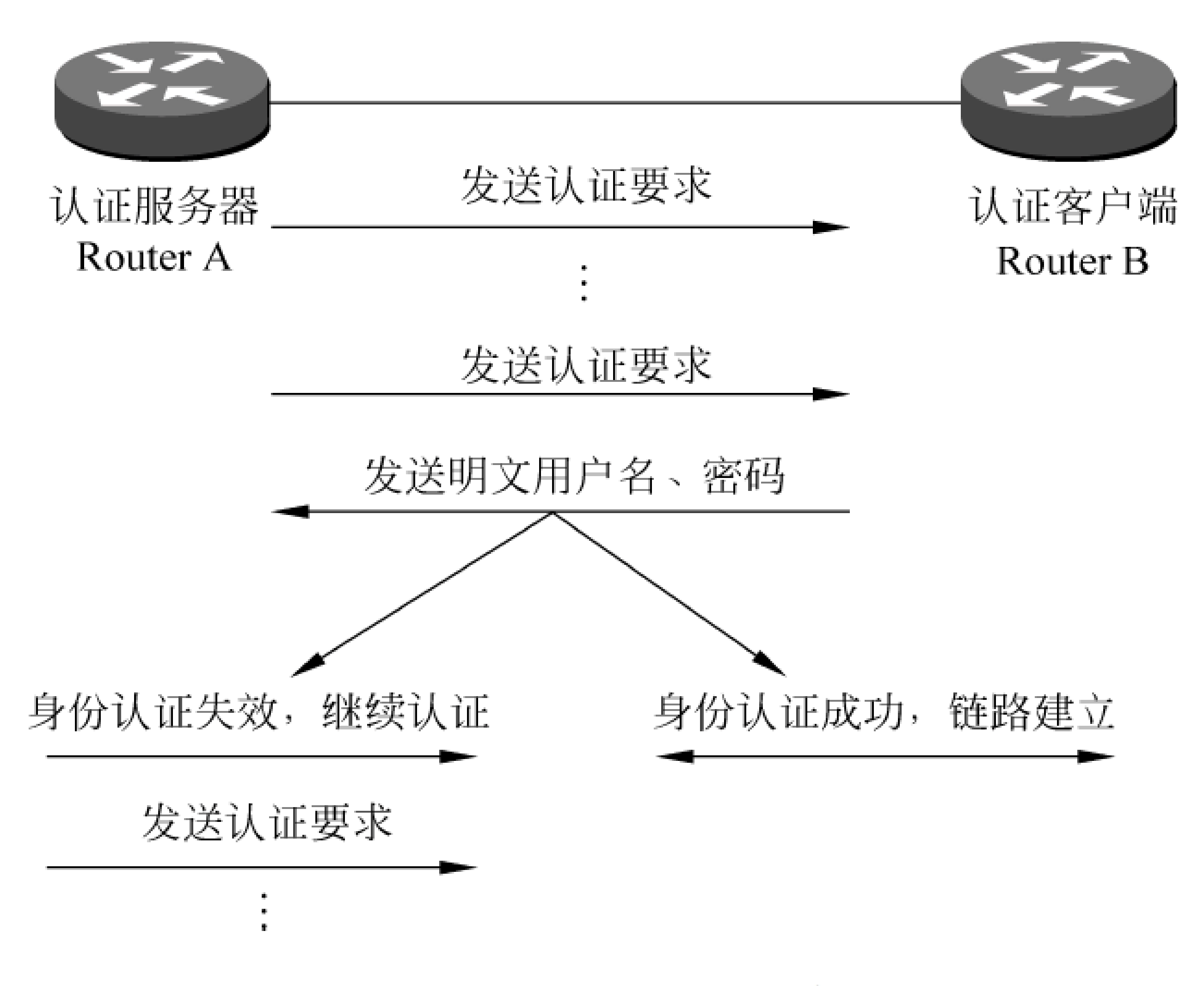


图 3.14 PAP 认证示例

在图 3.14 中，当认证客户端（被认证一端）路由器 Router B 发送了用户名和密码后，认证服务器会将收到的用户名与密码和本地数据库中的密码信息比对，如果正确则身份认证成功，通信双方的链路最终成功建立。

如果被认证一端路由器 Router B 发送了错误的用户名或密码，认证服务器将继续不断地发送身份认证要求直到收到正确的用户名和密码为止。

(1) PAP 认证服务器端配置

PAP 认证服务器的配置分为两个步骤：建立本地密码数据库、要求进行 PAP 认证。

① 建立本地密码数据库

通过全局模式下的命令 username username password password 来为本地密码数据库添加记录，如下所示。

```
RouterA(config)# username routerb password rapass
```

② 要求进行 PAP 认证

这需要在相应接口配置模式下使用命令 ppp authentication pap 来完成，如下所示。

```
RouterA(config)# interface serial 0/0
RouterA(config-if)# ppp authentication pap
```

(2) PAP 认证客户端配置

PAP 认证客户端的配置只需要一个步骤（命令），即将用户名和密码发送到对端，如下所示。

```
RouterB(config-if)# ppp pap sent-username routerb pass rapass
```

2. PPP CHAP 认证

挑战握手认证协议（CHAP）由 RFC 1994 定义。

(1) CHAP 的认证过程

挑战握手认证协议通过三次握手周期性的认证对端的身份，在初始链路建立时完成，可以在链路建立之后的任何时候重复进行。

CHAP 认证过程以及 PPP CHAP 的工作过程分别如图 3.15 和图 3.16 所示。本地路由器（被认证者）和远程访问路由器 NAS（认证者）之间使用 PPP 协议进行通信，并使用 CHAP 进行身份鉴别。在鉴别之前，双方数据库中保存和对方通信的共享密钥（secret），该密钥也可以是双方共享的密码字。

CHAP 的认证过程描述如下。

① 链路建立阶段结束之后，认证者向对端（被认证者）发送“挑战”消息。

② 对端采用双方的共享秘密密码作为输入，对“挑战”使用单向哈希函数计算出一个密文。

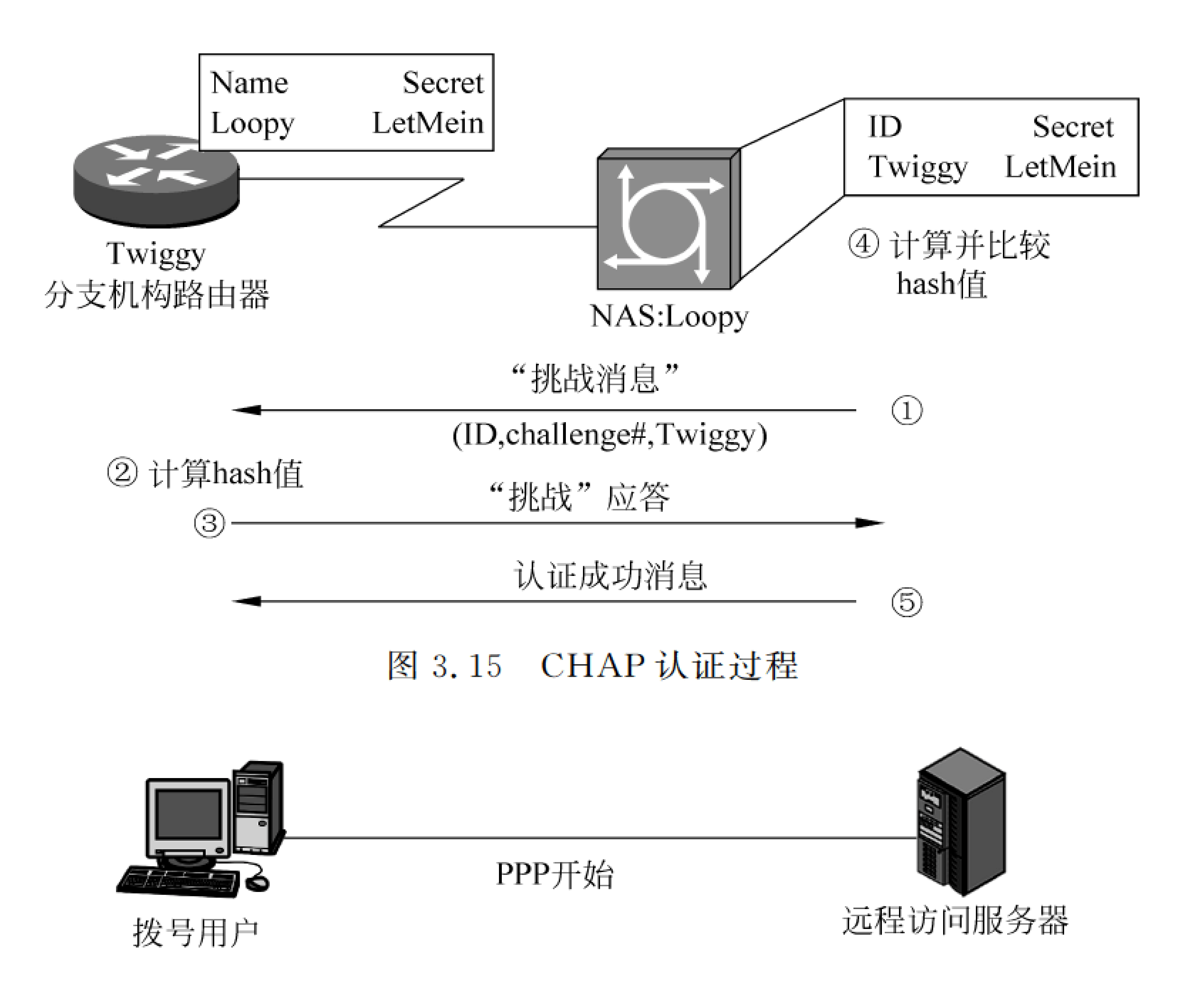


图 3.15 CHAP 认证过程

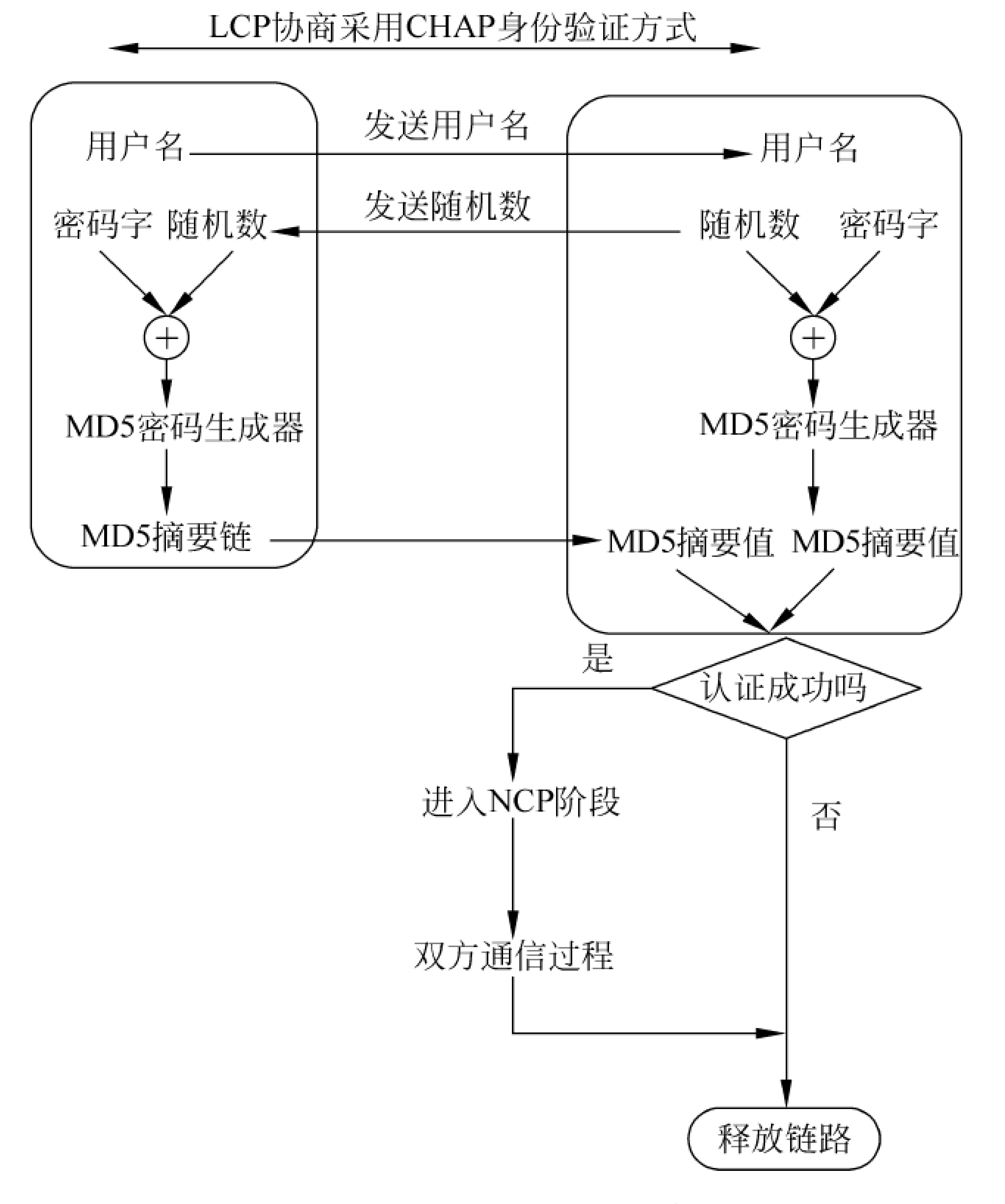


图 3.16 PPP CHAP 的工作过程

③ 对端将此密文经过网络发送至认证者，作为对"挑战"的应答。

④ 认证者按照同样的算法和输入计算一个自己期望的哈希值，通过检查该值和应答消息是否匹配来决定是否通过认证。

⑤ 通过认证后，向对端发送"认证成功消息"，并进入 PPP 协议的 NLP 阶段，否则，连接被终止。

经过一定的随机时间间隔，认证者向对端发送一个新的"挑战"，然后，重复上述的第①～⑤步进行下一轮的认证过程。

(2) CHAP 协议的安全性分析

CHAP 认证比 PAP 认证更安全，因为 CHAP 协议中的密码保存在认证对等端各自的数据库中，不在网络上传输。被认证端发送的只是经过摘要算法加工过的随机序列(挑战字符串的应答)。同时，在双方正常通信过程中，身份认证可以随时进行，而 PAP 中的鉴别只发生在链路建立阶段。通过递增改变的标识符和可变的挑战值，CHAP 可防止重放攻击，重复挑战限制了对单个攻击的暴露时间，认证者控制挑战的频度。

该认证方法依赖于认证者和对端共享的密钥，虽然该认证是单向的，但是在两个方向都进行 CHAP 协商，同一密钥可以很容易实现交互认证。

CHAP 算法要求密钥长度必须至少是一字节，至少应该不易让人猜出，密钥最好至少是哈希算法(例如 MD5 的 16 字节)所选用的哈希值的长度，如此可以保证密钥不易受到穷举攻击。所选用的哈希算法，必须使得从已知挑战值和应答值来确定密钥在计算上不可行。

每一个挑战值应该是唯一的，否则在同一密钥下，重复挑战值将使攻击者能够用以前截获的应答值响应挑战。由于希望同一密钥可以用于地理上分散的不同服务器的认证，因此挑战应该全局临时唯一。

每一个挑战值也应该是不可预计的，否则攻击者可以欺骗对端，让对端响应一个预计的挑战值，然后用该响应冒充对端欺骗认证者。

虽然 CHAP 不能防止实时的主动搭线窃听攻击，但只要能产生不可预计的挑战就可以防范大多数的主动攻击。

CHAP 对端系统要求很高，因为需要多次进行身份质询、响应。这需要耗费较多的 CPU 资源，因此只用在对安全要求很高的场合。

同 PAP 一样，CHAP 认证可以在一方进行，即由一方认证另一方身份，也可以进行双向身份认证。这时，要求被认证的双方都要通过对方的认证程序，否则，无法建立两者之间的链路。

(3) 算法协商和数据包格式

在使用 CHAP 进行验证之前，通信对等体之间需要使用 PPP 的"认证协议配置选项"(configuration option format)消息协商认证协议和认证算法。"认证协议配置选项"数据包格式如图 3.17 所示。

类型	长度	认证协议	算法

图 3.17 认证协议配置选项数据包格式

- 类型(type):表示认证协议类型,CHAP 为 3。
- 长度(length):固定为 5 个字节。
- 认证协议(authentication-protocol):对于 CHAP 为 0xc223(十六进制)。
- 算法(algorithm):算法字段一字节,指示所使用的认证方法,至少应实现如下算法——MD5 下的 CHAP。

CHAP 数据包的格式如图 3.18 所示。

代码	标识符	长度	数据

图 3.18 CHAP 数据包格式

CHAP 数据包封装在 PPP 数据链路层帧的信息域中,当 PPP 帧的协议字段是十六进制的 0xc223 时,表示 PPP 帧的信息字段里封装了一个完整的 CHAP 报文。CHAP 报文的格式描述如下。

- 代码(code)。代码字段,1 字节,指示 CHAP 报文的类型,分配如下。
  ——1:挑战(challenge)。
  ——2:应答(response)。
  ——3:成功(success)。
  ——4:失败(failure)。
- 标识符(identifier)。标识符字段,1 字节,辅助匹配挑战、应答和响应。
- 长度(length)。长度字段,2 字节,指示 CHAP 报文的长度,包括代码、长度和数据字段。超出长度的字节应该视为数据链路层填充,接收方应该忽略。
- 数据(data)。数据字段,0 个或多个字节,数据字段类型由代码字段确定。

挑战报文是 CHAP 的开始,认证者必须传送代码字段为 1 的 CHAP 报文,其他挑战报文必须在有效应答报文成功接收之后或重试计数器计满后发送。

为了确保连接没有被更改,挑战报文也可以在 NLP 阶段的任何时候发送。对端应该随时为认证阶段和 NLP 阶段的挑战做好准备,任何时候收到挑战报文,对端都必须传送 CHAP 应答报文。

无论何时,如果收到应答报文,认证者都必须把应答值和自己计算的预期值比较,基于这种比较,认证者必须发送成功(success)或者失败(failure)CHAP 报文。

(4) CHAP 认证实例

以单方认证为例分析 CHAP 协议的认证和配置过程。

如图 3.19 所示,两个路由器之间封装了 PPP 协议且要求进行 CHAP 身份认证,同时

它们之间的链路在物理层激活后，认证服务器(认证者)会不停地向认证客户端(被认证者)发送身份认证要求直到身份认证成功。和 PAP 不同的是，这时认证服务器发送的是“挑战”字符串。

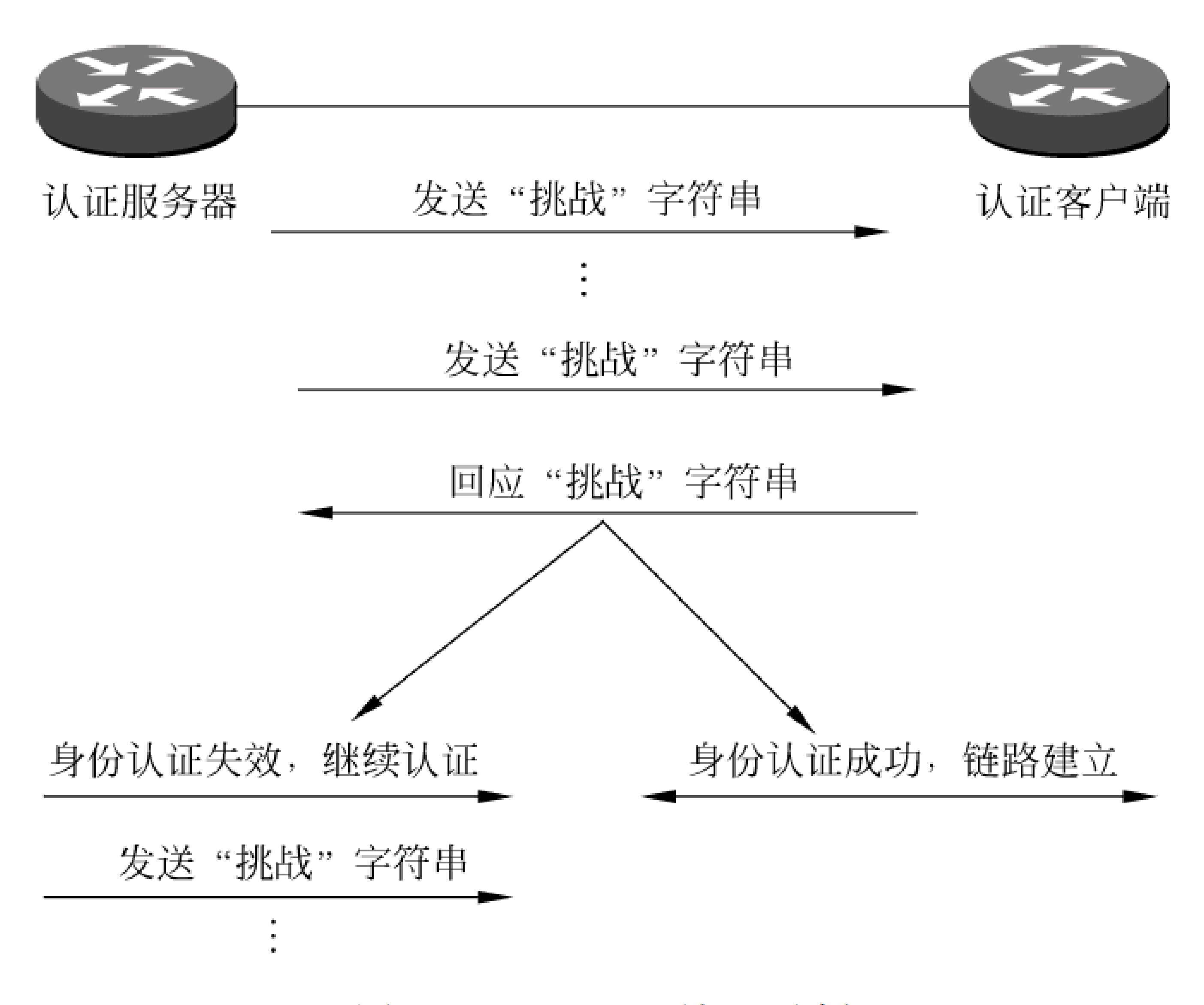


图 3.19 CHAP 认证示例

这里的“挑战字”包括会话 ID 和一个随机字符串(arbitrary challenge string)。远程客户使用 MD5 单向哈希算法(one-wayhashing algorithm)返回用户名和加密的挑战密码、会话 ID 以及用户密码，其中用户名以非哈希方式发送。即

```
Challenge = (Session ID,Challenge String)
Response = MD5(Session ID,Challenge String,User Password),User Name
```

当认证客户端路由器 Router B 发送了对“挑战”字符串的应答数据包后，认证服务器按照同样的算法和参数计算消息摘要以验证对方的身份。如果正确，则身份认证成功，通信双方的链路最终成功建立。

如果被认证方路由器 Router B 发送了错误的“挑战”应答数据包，认证服务器将继续不断地发送身份认证要求，直到收到正确的应答数据包为止。

① CHAP 认证服务器的配置。

CHAP 认证服务器的配置分为两个步骤：建立本地密码数据库，要求进行 CHAP 认证。

- 建立本地密码数据库。通过全局模式下的命令 username 和 password 为本地密码数据库添加记录。此处的 username 应该是对端路由器的名称，即 routerb，如下所示。

```
RouterA(config)# username routerb password samepass
```

- 要求进行 CHAP 认证。这需要在相应接口配置模式下使用命令 ppp authentication chap 来完成，如下所示。

```
RouterA(config)# interface serial 0/0
RouterA(config-if)# ppp authentication chap
```

② CHAP 认证客户端的配置。

CHAP 认证客户端的配置只需要一个步骤（命令），即建立本地密码数据库。此处的 username 应该是对端路由器的名称，即 routera，而密码应该和 CHAP 认证服务器密码数据库中的密码相同，如下所示。

```
RouterB(config-if)# username routera password samepass
```

CHAP 认证的缺点是要求密钥以明文形式存在，无法加密密码数据库。在大型设备中不适用，因为每个可能的密钥由链路的两端共同维护。

微软挑战—握手验证协议（MS-CHAP）是对 CHAP 的改进。同 CHAP 一样，使用 MS-CHAP 时，NAS 会向远程客户发送一个含有会话 ID 和任意生成的挑战字串的挑战密码。远程客户必须返回用户名以及经过哈希算法加密的挑战字串，会话 ID 和用户密码的哈希值。采用这种方式服务器端将只存储经过哈希算法加密的用户密码而不是明文密码，这样就能够提供进一步的安全保障。此外，MS-CHAP 同样支持附加的错误编码，包括密码过期编码以及允许用户自己修改密码的加密的客户—服务器（client-server）附加信息。使用 MS-CHAP，客户端和 NAS 双方各自生成一个用于随后数据加密的起始密钥。

3. PPP EAP 认证

PPP 扩展认证协议 EAP 由 RFC 2284 定义。

PPP 扩展认证协议 EAP 也可以用于 PPP 认证，它并不是一种具体的认证方法，而是一种认证机制，可以支持多种认证方法，包括一次性密码 OTP、挑战握手认证协议 CHAP 等。EAP 并不在链路控制阶段指定认证方法，而是把这个过程推迟到认证阶段。这样认证方就可以在要求更多的信息以后再决定使用什么认证方法。这种机制就允许使用一台“后端”服务器（back-end server）来真正执行认证机制，而 PPP EAP 认证方只是向该服务器传递认证交换信息。

EAP 协议的要点及工作过程描述如下。

① 在链路建立阶段完成以后，认证方向对端发送一个或多个请求报文去认证节点。在请求报文中有一个类型字段用来指明认证方所请求的信息，该字段实际上即对应不同的认证方法，例如是 ID、MD5 的挑战字（PPP CHAP）、一次密码（OTP）以及通用令牌卡等。MD5 的挑战字对应于 CHAP 认证协议的挑战字。通常认证方首先发送一个初始的 ID 请求随后再发送其他的请求信息。当然，这个 ID 请求报文并不是必需的，在对端身份是已知的情况下（如租用线、拨号专线等）可以跳过这个步骤。

② 端点对每一个请求报文回应一个应答包。和请求报文一样，应答报文中也包含一个类型字段，对应于所回应的请求报文中的类型字段。

③ 认证方通过发送一个成功或者失败的报文来结束认证过程。

和 CHAP 相同，EAP 也使用 PPP 的“认证协议配置选项”消息协商认证协议（如图 3.18 所示），对于 PPP 中的 EAP，认证协议字段（authentication-protocol）为 C227（十六进制）。EAP 数据包格式和 CHAP 类似（如图 3.20 所示）。数据包类型也分为请求、应答、成功和失败 4 种。

其中，请求和应答数据包的格式如图 3.20 所示。类型字段类型（type）占一个字节。这个字段表示请求或者应答的信息类型。每一种 EAP 请求或者应答报文必须指定并且也只能指定一种类型。该类型实际对应一种认证方法。一般情况下，应答报文中的类型字段和请求报文中的类型字段是相同的，但是还存在一种为 NAK 的应答类型用来表示对端不接受请求报文中的信息类型。当对端用 NAK 报文应答请求报文的时候，对端可以同时提供一个它所支持的信息类型供认证者选择。

代码	标识符	长度	类型	数据

图 3.20　EAP 数据包格式

类型字段包括如下几种。

- 标识（identity）。
- 通知（notification）。
- 否定 NAK（只用在应答报文中）。
- MD5 挑战字（MD5-challenge）。
- 一次密码（one-time password，OTP）。
- 通用令牌卡（generic token card）。

其中，MD5 挑战字和 PPP 的 CHAP 协议中的挑战字类似，使用 MD5 算法。在类型为 MD5 挑战字的请求报文中包含一个“挑战”信息，对端收到这个请求报文后必须发送一个应答报文，应答报文的信息类型可以是 4（MD5 挑战字）或者 3（否定），在 NAK 应答报文中对端同时也标明了它所期待的认证机制的类型值。所有的 EAP 实现必须支持 MD5 挑战字算法。

字段类型为 5 时，请求报文中包含一个可显示的信息作为一次密码（OTP）的挑战字，说明双方采用 OTP 作为实际的身份认证方法。对端收到这种请求报文后必须发送一个应答报文，应答报文的类型值也必须设为 5（OTP）或者 3（NAK），在 NAK 应答报文中对端同时也标明了它所期待的认证机制的类型值。同时，在请求报文中，类型数据字段（type-data）包含一个可显示的信息作为一次密码（OTP）的挑战字。在应答报文中类型数据字段用于填充从 OTP 目录中得到的一次性密码。

字段类型为 6 时，代表使用通用令牌卡进行身份鉴别。该鉴别方法适用于各种需要用户输入信息的令牌卡的实现。在请求报文中包含一段 ASCII 文本信息，而应答报文中包含用于认证的令牌卡信息。典型的，令牌卡信息由用户从令牌卡设备上读取得到并作为 ASCII 文本输入。在该请求报文中，类型数据字段包含一段长度大于零的可显示的信息，它的长度可以从报文的长度字段中计算得到。对端收到这种请求报文以后必须发送一个类型值为 6(通用令牌卡)的报文作为应答，应答报文中包含用于认证的令牌卡信息，其长度也可以从报文的长度字段中计算得到。

图 3.21 中给出了一个使用 EAP 进行身份鉴别的例子。

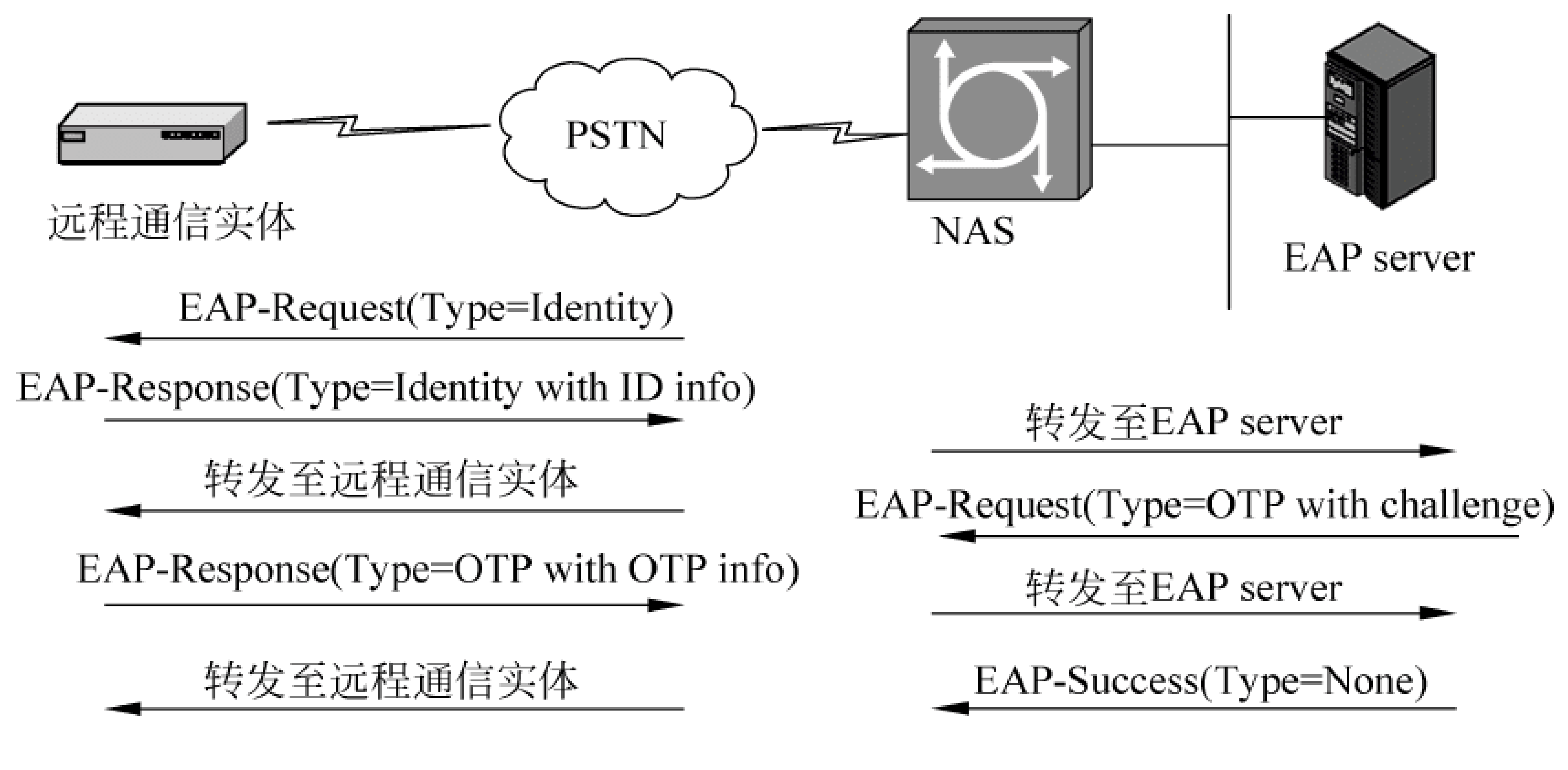


图 3.21 PPP EAP 认证过程

EAP 的优点是可以支持多种认证机制，而无须在 LCP 阶段预协商过程中指定。某些设备(如网络接入服务器 NAS)不需要关心每一个请求报文的真正含义，而是作为一个代理把认证报文直接传输给后端的认证服务器。设备只需关心认证结果是成功还是失败，然后结束认证阶段。并且，由于使用专门的后端服务器进行验证，使得远程访问服务器 RAS 在验证系统升级后不需要更换。

EAP 的缺点是 EAP 需要在 LCP 中增加一个新的认证协议，这样现有的 PPP 实现要想使用 EAP 就必须进行修改。同时，使用 EAP 也和现有的在 LCP 协商阶段指定认证方法的模型不一致，因为它不在链路控制阶段指定认证方法，而是把这个过程推迟到认证阶段由 EAP 协议来确定。

3.5.2 AAA 协议及其应用

认证、授权和记账(authentication，authorization，accounting，AAA)，它在鉴别的同时提供授权和计费功能。其典型的应用实例包括 TACACS+和 RADIUS 协议。

实施 AAA 的网络拓扑如图 3.22 所示。系统中有用户、AAA 客户端和 AAA 服务器三个角色。用户通过公共网络(例如图中的拨号网络或 Internet)访问远程服务器(network

access server,NAS)。NAS 可以是拨号服务器、VPN 服务器或无线访问点,NAS 作为 AAA 的客户端,接收用户信息(例如用户密码字或对挑战的应答)并将用户信息发送至 AAA 服务器,即 RADIUS 或 RADIUS 服务器。它们作为 AAA 的服务器端,提供身份认证、授权和记账服务。

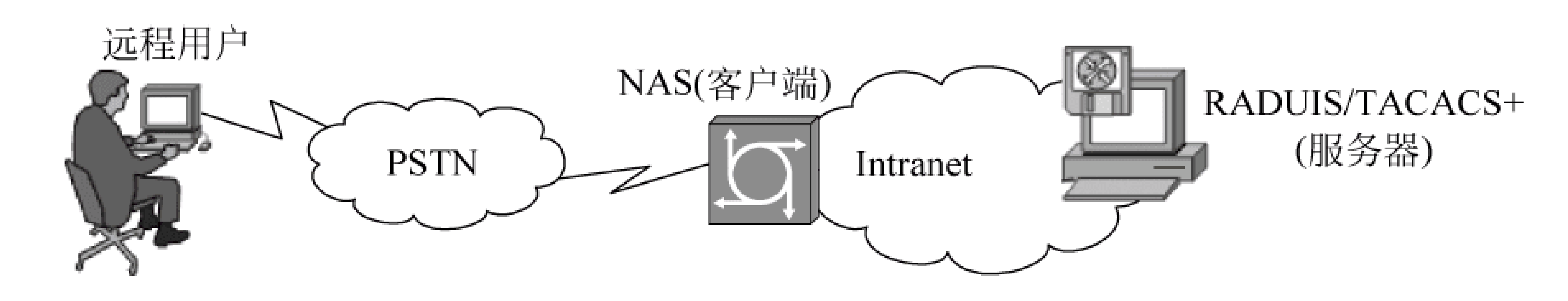


图 3.22　实施 AAA 的网络拓扑

1. RADIUS

RADIUS 由 RFC 2865"远程身份验证拨入用户服务(RADIUS)"和 RFC 2866"RADIUS 记账"定义。

RADIUS(remote authentication dial in user service)协议最初是由 Livingston 公司提出的,最初是为拨号用户进行认证和计费设计的,后来经过多次改进,形成了一项通用的认证计费协议。RADIUS 认证要用到基于挑战/应答(challenge/response)的认证方式。

RADIUS 是一种 C/S 结构的协议,它的客户端最初就是 NAS(net access server)服务器,现在任何运行 RADIUS 客户端软件的计算机都可以成为 RADIUS 的客户端。RADIUS 协议认证机制灵活,可以采用 PAP、CHAP 或者 UNIX 登录认证等多种方式。RADIUS 是一种可扩展的协议,它进行的全部工作都是基于 Attribute-Length-Value 的向量进行的。

RADIUS 的基本工作原理是用户接入 NAS,NAS 向 RADIUS 服务器使用 Access-Require 数据包提交用户信息,包括用户名、密码等相关信息,其中用户密码是经过 MD5 加密的,双方使用共享密钥,这个密钥不经过网络传播;RADIUS 服务器对用户名和密码的合法性进行检验,必要时可以提出一个 Challenge,要求进一步对用户认证,也可以对 NAS 进行类似的认证;如果合法,给 NAS 返回 Access-Accept 数据包,允许用户进行下一步工作,否则返回 Access-Reject 数据包,拒绝用户访问;如果允许访问,NAS 向 RADIUS 服务器提出计费请求 Account-Require,RADIUS 服务器响应 Account-Accept,对用户的计费开始,同时用户可以进行自己的相关操作。

RADIUS 服务器支持各种用户身份认证方法,如 PPP、密码验证协议(PAP)、质询握手验证协议(CHAP)、Unix 登录及其他认证机制。

RADIUS 协议中,验证和授权是组合在一起的。如果验证通过,RADIUS 服务器将返回一个 Access-Accept 响应,其中包括一些参数(属性-值对),以保证对该用户的访问。这些参数是在 RADIUS 中配置的,包括访问类型、协议类型、IP 地址以及一个访问控制列表

(ACL)或要在 NAS 上应用的静态路由等。

RADIUS 服务器和 NAS 服务器通过 UDP 协议进行通信,RADIUS 服务器的 1812 端口负责认证,1813 端口负责计费工作。采用 UDP 的基本考虑是因为 NAS 和 RADIUS 服务器大多在同一个局域网中,使用 UDP 更加快捷方便。

典型的通信过程如图 3.23 所示。远程用户(user)使用 PPP 协议访问拨号服务器(NAS),NAS 是 RADIUS 客户端,用户信息存放在 RADIUS 服务器中。NAS 和 RADIUS 服务器之间拥有事先协商产生的共享密钥(shared secret)。其通信过程描述如下。

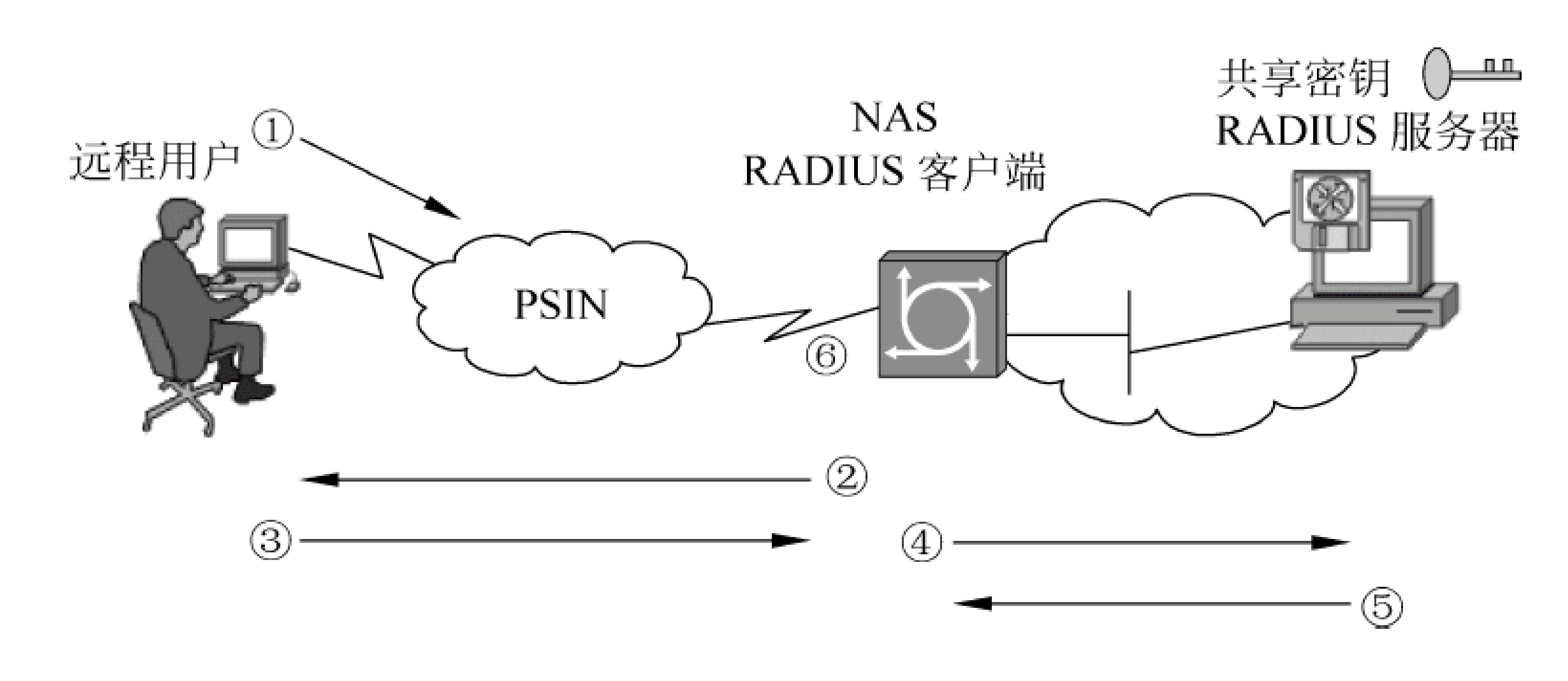


图 3.23 RADIUS 通信过程

① 远程用户向 NAS 发起 PPP 身份认证请求,在请求报文中指明认证方式,例如采用 PAP 或 CHAP 等。该报文不是 RADIU 协议数据包,而是封装在 PPP 报文中的鉴别请求消息,通信过程类似 CHAP 和 PAP。

② NAS 根据具体认证方式向远程服务器进行应答。例如,若是 PAP 认证,则提示远程用户输入用户名和密码(PAP ID/PASSWORD);若是 CHAP 则向远程用户发送一个"挑战字"(challenge)。

③ 远程用户向 NAS 产生应答。例如,若是 PAP 则发送自己的用户名和密码(PAP ID/PASSWORD);若是 CHAP 则发送一个"挑战"应答(challenge response)。

④ NAS 将从③中收到的远程用户的密码(或挑战应答),以及远程用户的用户名等封装成 RADIUS 消息(ACCESS-REQUEST 消息),发送至 RADIUS 服务器进行身份鉴别和授权请求。这里,NAS 将在 ACCESS-REQUEST 消息中加密远程用户的密码(例如针对 PAP 的明文密码)。加密时使用 MD5 算法,并使用 NAS 和 RADIUS 服务器之间的共享密钥(shared secret)作为参数。这样做的目的是防止密码明文传输,同时 RADIUS 服务器也可以鉴别 NAS 的身份(通过双方的共享密钥)。

⑤ RADIUS 收到该 ACCESS-REQUEST 消息后,根据自己数据库中保存的用户信息和密码字对远程用户进行身份鉴别(若是 PAP 协议中加密后的密码则采用同样的算法计算摘要值进行比对;若是 CHAP 中的挑战应答消息,则根据 CHAP 算法计算消息摘要值),决定鉴别和授权结果。然后将鉴别结果封装为 RADIUS 消息(access accept 或 access reject)

发送至 NAS。

⑥ NAS 根据该鉴别和授权结果，决定向远程用户提供何种服务。

在 NAS 和 RADIUS 服务器之间通信的 RADIUS 协议的消息格式如图 3.24 所示。RADIUS 报文封装在 UDP 报文的数据域中，它的 UDP 目的端口号是 1812。

代码域	标识符	长度
认证字		
属性		

图 3.24 RADIUS 消息格式

- 代码域（code），标识 RADIUS 消息的类型，如 ACCESS-REQUEST、ACCESS-ACCEPT、ACCESS-REJECT、计费请求和计费应答等。
- 标识符（identifier），用于匹配请求和回应报文。如果在一个很短的时间内接收到相同的源 IP 地址、源 UDP 端口号和相同的 Identifier 域的请求报文，RADIUS 服务器就可以认为是重复的请求报文。
- 长度域，包括 Code 域、Identifier 域、Length 域、Authenticator 域和属性域在内的总长度。如果包的实际长度小于长度域中给出的值，该包必须被静默丢弃。报文的最小长度是 20 字节，最大长度是 4096 字节。
- 认证字（authenticator），该域的值用来鉴别服务器的应答报文，并且用在用户密码的隐藏算法中。包括请求认证字、应答认证字、管理提示等。
- 属性字段，针对不同报文具有不同取值，例如对于 ACCESS-REQUEST，该字段包括用户名和密码等信息。

例如，远程用户使用 PAP 身份认证方式，用户名是 nemo，NAS 的 IP 地址为 192.168.0.16，端口号为 3。此时，NAS 获得了用户名和密码（password），则在上述第 4 步中，它向 RADIUS 服务器发送的 access-request 消息格式如下。

```
Code = 1(Access - Request)
ID = 0
Length = 56
Request Authenticator(NAS 产生的 16 字节的随机数)
```

属性字段（Attributes）包括：

```
User - Name = "nemo"
User - Password
NAS - IP - Address = 192.168.0.16
NAS - Port = 3
```

其中，加密的密码（user-password）的值通过如下方法计算。

① 将从远程用户获得的密码字（password）填充至 16 字节，得到 Padded-password。

② 产生 16 字节的随机数将该随机数和共享密钥（shared password）一起计算 MD5 消息摘要，即 Hash1＝MD5（random＃，secret）。

③ 将以上结果进行异或运算，得到加密密码，携带在属性字段的 User-password 中传

递至服务器端。即 User-password＝hash1 XOR Padded-password。

由于加密密码计算中使用的随机数将携带在 Request Authenticator 中传递至服务器，服务器可以采用相同的算法计算出该消息摘要，并和加密密码进行比对，根据比对结果决定是否通过对远程用户的身份认证。

对于 CHAP 认证方式，NAS 从远程用户处得到的是一个挑战应答，该值是经过 CHAP 加密后的消息摘要。这种情况下，NAS 直接将该挑战应答以及用户名和 CHAP ID 等值传递给服务器，不再对该挑战应答进行加密。服务器根据用户名从其数据库中取得用户密码，并采用 CHAP 算法计算消息摘要进行比对。

RADIUS 协议应用范围很广，包括普通电话、上网业务计费、VPN 服务中根据不同用户分配不同权限、无线宽带接入认证与计费等。Windows Server 的身份验证服务器 IAS 即可以配置为 RADIUS 服务器。RAS 和 IAS 配合可以为 VPN 和 PPP 拨号用户提供 AAA 服务。另外，有许多开源的 RADIUS 服务器可以配置在 Linux 或 UNIX 系统中，例如，FreeRADIUS 可以安装并配置在 Linux 下。

下面讲述一个使用 RADIUS 进行 AAA 的应用示例，该示例中，通过 AAA 协议的实施可以完成如下用户需求。

- 用户通过拨号验证访问公司内部网。
- 带无线支持的笔记本电脑可以通过无线验证连接到校园网。
- 管理员使用他们的工作站安全登录到网络设备上进行网管配置。

所有这些验证任务都通过 RADIUS 服务器，并基于一个中央 LDAP 服务器来完成。

网络拓扑如图 3.25 所示。NAS 提供拨号服务，WAP 提供无线接入服务，由 LDAP 服务器完成实际的验证过程。NAS 和 WAP 均采用硬件设备实现(如 Cisco 路由器和 Cisco

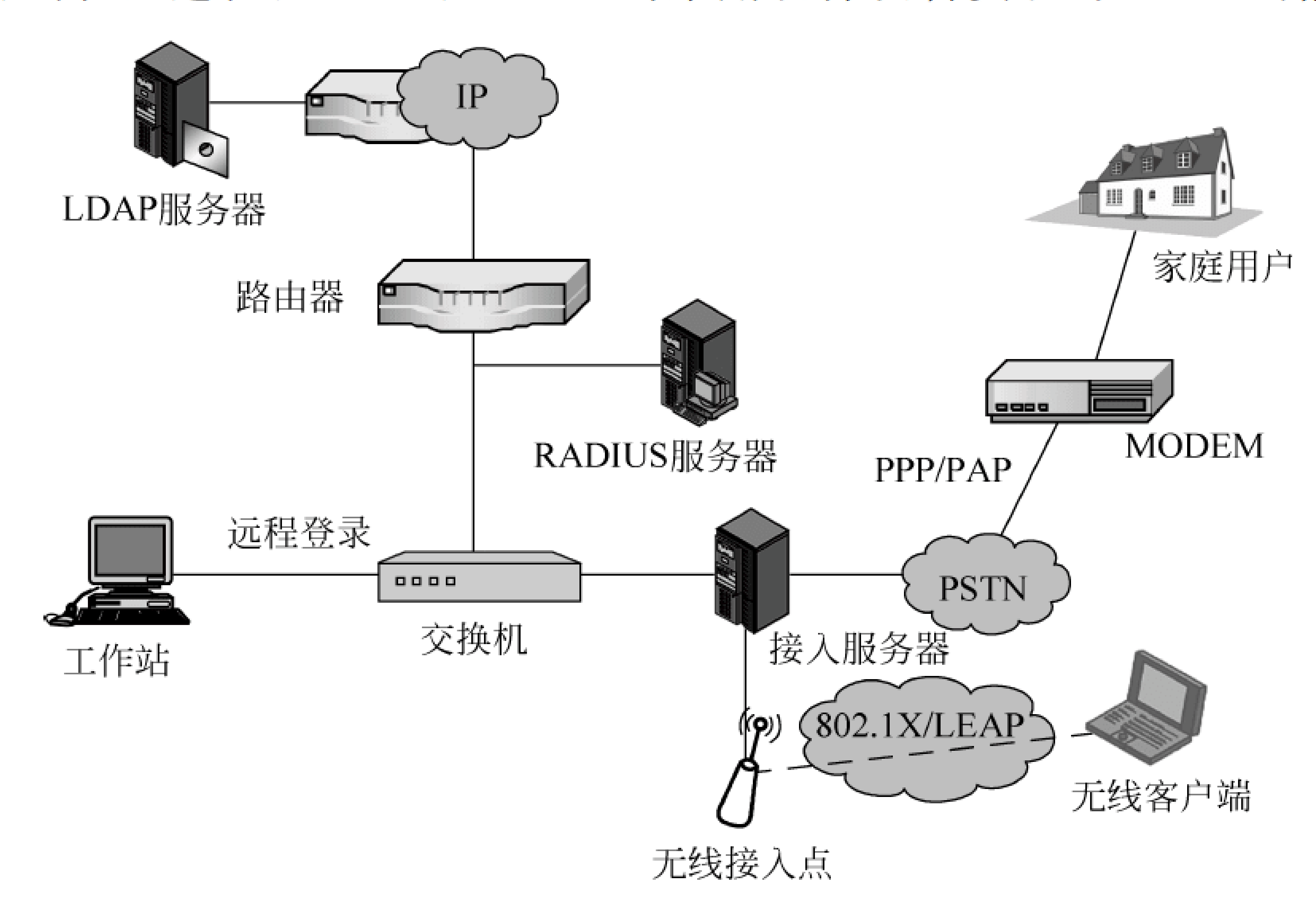


图 3.25　通过 RADIUS 和 LDAP 实施 AAA 示例

1200 series AP)，通过安装在 Linux 系统中的 FreeRADIUS 实现 RADIUS 服务器。图中的 NAS 也可以使用 Windows 的远程访问服务器 RAS(Remote Access Server)代替。

FreeRADIUS 是开放源码的一种 Linux 下的 RADIUS 服务器，可用于分布式和异构计算环境。FreeRADIUS 支持 LDAP、MySQL、PostgreSQL 和 Oracle 数据库，并与 EAP 和 Cisco LEAP 等网络协议兼容。FreeRADIUS 目前被部署在很多大型生产网络系统中。

为了实现 AAA 功能，需要配置 RADIUS 服务器以及 NAS 和 WAP。

(1) 服务器端配置

RADIUS 服务器的配置在 Linux 服务器上进行，包括对服务器、客户端(NAS 和 WAP)和用户的配置。

- 首先要将服务器配置为使用 LDAP 进行身份认证。
- 客户机主要配置 RADIUS 服务器和 NAS 客户端之间的共享密钥(密码字)，如：

```
client 192.168.0.1 {
  secret       = mysecret1
  shortname    = myserver
  nastype      = other
}
```

- 为验证和授权配置用户信息。

(2) NAS 客户端配置

在网络访问服务器中需要指明 RADIUS 服务器的 IP 地址以及服务器的共享密钥，然后还需要配置使用 RADIUS 进行验证、授权和记账。

```
aaa new - model
radius - server host 192.168.0.100
radius - server key mysecret1
```

(3) WAP 客户端配置

WAP 配置类似于 NAS，需要指明：

- 服务器名或 IP 地址和共享的密钥。
- 选择 Radius 作为验证类型。

2. CISCO TACACS+

如图 3.26 所示，TACACS+的通信过程描述如下。

① 用户向 PPP 发出认证请求。

② NAS 发送 START 包到 TASACS+Server。

③ TASACS+Server 回应 GETUSER 包，包含 username/password(PAP)或 challenge (CHAP)的提示信息。

④ NAS 将响应显示给用户。

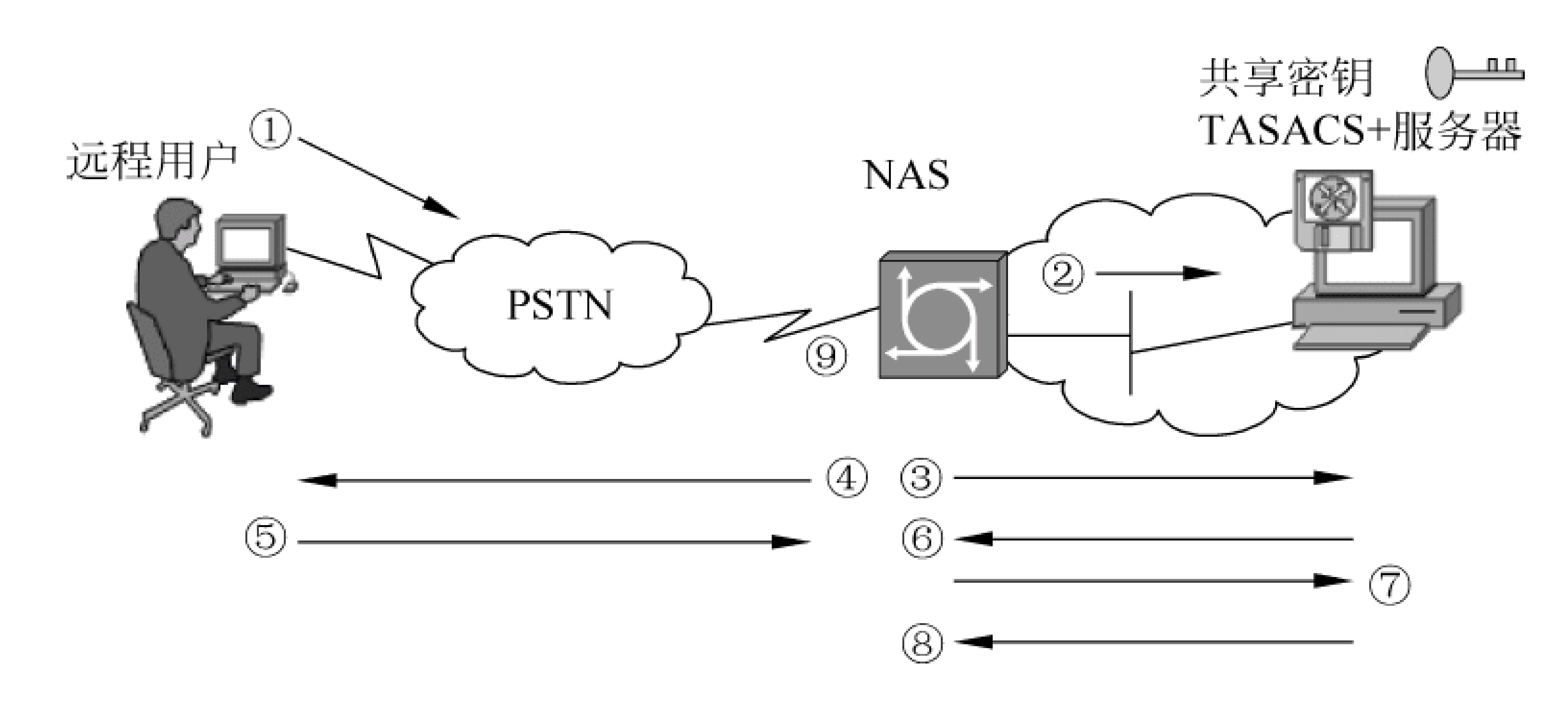


图 3.26 TACACS+通信过程

⑤ 用户回应 NAS。

⑥ NAS 发送加密的包到 TASACS+Server。

⑦ TASACS+Server 回应 NAS 认证结果。

⑧ NAS 和 TASACS+Server 交换授权信息和应答。

⑨ NAS 根据所交换的授权信息作出反应。

和 RADIUS 协议类似，TACACS+在客户机和服务器之间具有共享密钥，通过该共享密钥互相鉴别，并加密通信过程。其中，密码计算过程描述如下。

① Hash1 = MD5(session ID，secret，version#，seq#)。

② Hash2 = MD5(hash1，session ID，version#，seq#)。

③ 重复指定次数。

④ 最后一个散列值被填充或截取到和被加密数据相同的长度，该过程称为伪填充(pseudo-pad)。

⑤ 将数据和伪填充的值进行异或得到密文。此密文在客户端(NAS)和 TACACS+服务器之间传输。

TACACS+ 和 RADIUS 的比较如下。

• TACACS+使用 TCP 协议，而 RADIUS 使用 UDP 协议。

• RADIUS 为工业标准，并由 RFC 定义，而 TACACS+ 是 Cisco 特有的。

• 在客户端和 RADIUS 服务器之间的消息，RADIUS 只加密密码，而 TACACS+则加密整个数据包。

3.5.3 Kerberos 鉴别

Kerberos 协议是为 TCP/IP 网络设计的、以可信第三方为基础的认证协议，最初由 MIT 开发，它基于 Needham-Sehroeder 协议，并在 Needham-Schroeder 协议中引入了时间戳处理机制，使用的是对称密钥体系。

Kerberos 建立了一个中心认证服务器 KDC 向用户和服务器提供相互认证,保证只有通过认证的用户才能访问服务器,以防止未授权访问。Kerberos 采用对称密钥体制(支持 DES 算法,也可用其他算法代替)对信息进行加密。其基本思想是:用户在对应用服务器进行访问之前,必须先从第三方(Kerberos 服务器)获取该应用服务器的访问许可证(即票据,ticket)。

目前该协议已经有 5 个版本,其中 V1 到 V3 是内部开发版,V4 是 1988 年开发的,而 V5 对 V4 中的某些安全缺陷做了改进,并于 1994 年作为 RFC 标准公布。目前,Kerberos 协议已被广泛应用于多种操作系统,如 Windows、FreeBSD 中用来进行身份认证。

Kerberos 的设计是针对如下安全需求提出的。

- 安全性:网络中的窃听者不能获得必要的信息以假冒合法用户。
- 可靠性:对基于 Kerberos 访问的所有服务来说,Kerberos 系统的瘫痪意味着它所支持的所有服务的瘫痪。因此 Kerberos 服务应该是高度可靠的,应使用一个分布式服务结构实现该服务。
- 透明性:认证过程对用户透明,用户访问系统时,只需要输入原始密码。
- 可扩展性:系统应能够支持多种用户和应用服务器。

Kerberos 的设计基于以下前提实现。

- 用户必须在会话开始时向服务器证明自己的身份。
- 用户的密码不能以明文形式在网络中传输。
- 每个用户和服务器之间都拥有访问系统所使用的密码。
- 只有认证服务器(authentication server,AS)拥有所有用户和服务器的密码。

Kerberos 协议中使用的符号说明如下。

- C:客户端(client)。
- KDC:票据分发中心,包括 AS。
- AS:认证服务器(authentication server)。
- TGS:票据分发服务器(ticket granting server)。
- S:应用服务器(server)。
- TS_i:第 i 个时间戳。
- $Lifetime_i$:第 i 个有效生存期限。
- Addr:客户端 C 的 IP 地址。
- $Authenticator_i$:第 i 个认证符。
- K_C:客户端 C 的密钥。
- K_{tgs}:tgs 的密钥。
- K_S:应用服务器 S 的密钥。
- $K_{C,tgs}$:客户端 C 和 tgs 共享的会话密钥。
- $K_{C,s}$:客户端 C 和应用服务器 S 共享的会话密钥。

- TGT：用于访问 TGS 的票据。
- T_S：用于访问应用服务器 S 的票据。
- $\{M\}K_X$：用密钥 K_X 对报文 M 进行加密。

说明：在 AS 服务器中，保存有 K_C，K_{tgs} 和 K_S 密钥，AS 服务器分别与客户端、TGS 服务器和应用服务器共享这些密钥。

如图 3.27 所示，客户端 C 要访问应用服务器 S，需要进行 6 次协议交换，协议交换过程如下。

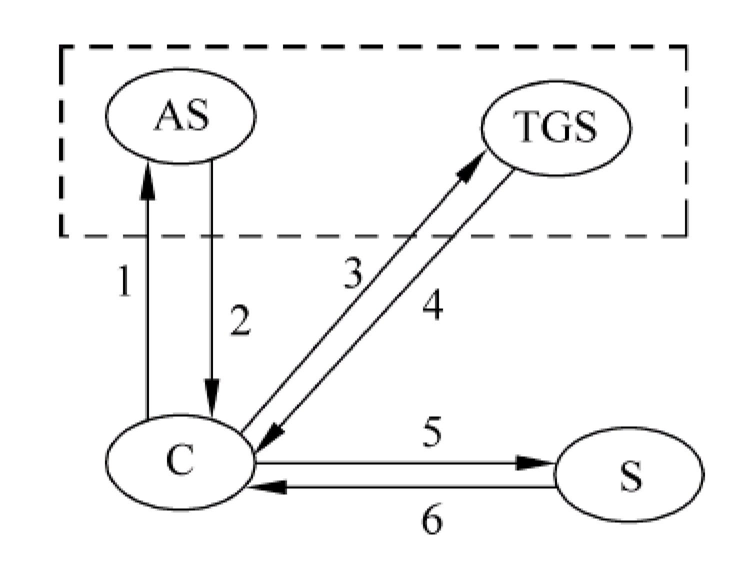


图 3.27　Kerberos 认证流程图

第一阶段(AS 交换)：客户端从 AS 处获取 TGT。

(1) C－＞AS：C，TGS，Addr，TS_1

客户端向 AS 发出访问 TGS 的请求，请求报文包括客户端的名字、TGS 的名字、客户端的 IP 地址以及时间戳。时间戳 TS_1 用于向 AS 表示这一请求是新的。请求报文以明文方式发送。

(2) AS－＞C：$\{K_{C,tgs}, TGT\}K_C$

其中，TGT 为 $\{TGS, C, Addr, TS_2, Lifetime_2, K_{C,tgs}\}K_{tgs}$。

AS 收到客户端请求报文后，产生随机会话密钥 $K_{C,tgs}$ 和 TGS 的票据 TGT，用客户端的密钥 K_C 加密后作为应答报文。会话密钥 $K_{C,tgs}$ 用于客户端和 TGS 之间进行加密通信。TGT 的内容包括：TGS 的名字、客户端的名字、客户端的 IP 地址、时间戳、有效生存期限，以及会话密钥 $K_{C,tgs}$，这些数据使用 TGS 的密钥 K_{tgs} 进行加密，以保证只有 TGS 才能解密该密文。

AS 向客户端发出应答，应答内容使用客户端的密钥 K_C 加密，使得只有客户端 C 才能解密该报文的内容，即通过解密该密文，AS 可以鉴别客户端的身份。

客户端收到 AS 返回的应答报文后，用自己的密钥 K_C 进行解密，得到 TGS 的票据 TGT，客户端在下一步就可以把 TGT 发送给 TGS 来证明自己具有访问 TGS 的合法身份。客户端同时从 AS 处得到自己与 TGS 的会话密钥 $K_{C,tgs}$，用它来与 TGS 进行加密通信。

第二阶段(TGS 交换)：客户端从 TGS 处获取访问应用服务器的票据 T_S。

(3) C－＞TGS：S，TGT，$Authenticator_1$

其中，$Authenticator_1$ 为 $\{C, Addr, TS_3\}K_{C,tgs}$。

客户端向 TGS 发送访问应用服务器 S 的请求报文，报文内容包括要访问的应用服务器 S 的名字，TGS 的票据 TGT 以及认证符。

TGT 的内容用 TGS 的密钥 K_{tgs} 加密，只有 TGS 才能解开该密文。

认证符的内容包括客户端的名字、客户端的 IP 地址以及时间戳，认证符的内容用客户端和 TGS 的会话密钥进行加密，以保证只有 TGS 才能解密。

票据 TGT 可以重复使用且有效期较长，而认证符只能使用一次而且有效期很短。

TGS 收到客户端发来的请求报文后，用自己的密钥 K_{tgs} 对票据 TGT 进行解密处理，得

知客户端 C 已经从 AS 处得到与自己的会话密钥 $K_{C,tgs}$，此处票据 TGT 的含义为“使用 $K_{C,tgs}$的客户端是 C”。TGS 用 $K_{C,tgs}$解密认证符，并将认证符中的数据与 TGT 中的数据进行比较，从而可以相信 TGT 的发送者 C 就是 TGT 的实际持有者。

说明：此处的票据 TGT 并不能证明任何人的身份，只是用来安全地分配密钥，而认证符则用来证明客户端的身份。因为认证符只能被使用一次而且其有效期很短，所以可以防御针对票据和认证符的盗用。

（4）TGS－＞C：$\{K_{C,S}, T_S\}K_{C,tgs}$

其中，T_S 是用于访问应用服务器 S 的票据，内容为$\{S, C, Addr, TS_4, Lifetime_4, K_{C,S}\}K_S$。

TGS 检验客户端的合法身份之后，产生随机会话密钥 $K_{C,S}$，该密钥用于客户端 C 和应用服务器 S 进行加密通信，同时产生用于访问应用服务器 S 的票据 T_S，T_S 的内容包括：应用服务器的名字、客户端的名字、客户端的 IP 地址、时间戳、有效生存期和会话密钥 $K_{C,S}$，T_S 的内容用应用服务器 S 的密钥 K_S 加密，以保证只有 S 才能解密。会话密钥 $K_{C,S}$和票据 T_S 组成 TGS 的应答报文，该应答报文用客户端 C 和 TGS 的会话密钥 $K_{C,tgs}$加密，以保证只有客户端 C 才能解密。TGS 将该应答报文发送给客户端 C。

客户端 C 收到 TGS 的应答报文后，用会话密钥 $K_{C,tgs}$对报文进行解密，可以得到访问应用服务器 S 的票据 T_S，以及与 S 进行加密通信的会话密钥 $K_{C,S}$。只有合法用户 C 才能解密该报文的内容。

第三阶段（客户端—服务器认证交换）：客户端和服务器相互验证身份。

（5）C－＞S：S，T_S，$Authenticator_2$

其中，$Authenticator_2$ 的内容为$\{C, Addr, TS_5\}K_{C,S}$。

客户端 C 向应用服务器 S 发送请求报文，报文的内容包括应用服务器的名字，用于访问应用服务器 S 的票据 T_S 以及认证符。

T_S 的内容是用应用服务器 S 的密钥 K_S 加密的，只有 S 才能解密。

认证符的内容包括客户端的名字、客户端的 IP 地址、时间戳、认证符的内容用客户端和应用服务器的会话密钥加密，以保证只有应用服务器 S 才能解密。票据 T_S 可以重复使用且有效期较长，而认证符只能使用一次而且有效期很短。

应用服务器 S 收到客户端发来的请求报文后，用自己的密钥 K_S 对票据 T_S 进行解密处理，得知客户端 C 已经从 TGS 处得到与自己的会话密钥 $K_{C,S}$，此处票据 T_S 的含义为“使用 $K_{C,S}$的客户端是 C”。S 用 $K_{C,S}$解密认证符，并将认证符中的数据与 T_S 中的数据进行比较，从而可以相信 T_S 的发送者 C 就是 T_S 的实际持有者，客户端 C 的身份得到了验证。

（6）S－＞C：$\{TS_5+1\}K_{C,S}$

应用服务器 S 检验认为客户端 C 身份合法之后，对从认证符中得到的时间戳 TS_5 加 1，然后用与客户端 C 共享的会话密钥 $K_{C,S}$加密后作为应答报文发给客户端。该应答报文只有客户端 C 才能解密。

客户端 C 收到应用服务器 S 发来的应答报文后，用会话密钥 $K_{C,S}$进行解密后，对应答

报文中增加的时间戳进行验证,验证通过后,应用服务器 S 的身份也得到了验证。

整个协议交换过程结束以后,客户端和应用服务器之间就拥有了共享的会话密钥,双方随后可以用该会话密钥来进行加密通信或者交换新的会话密钥。

3.5.4 S/KEY 一次性密码鉴别

如前所述,防止密码猜测和字典攻击的方法是使用 OTP 一次性密码鉴别技术。目前互联网上普遍使用 Bellcore 的 S/KEY 一次性密码系统对付字典攻击。

S/KEY 由 IETF RFC 2289、RFC 2243 和 RFC 2444 等定义。

S/KEY 系统使用一个秘密的通行词(密码字)产生一系列一次性密码。

系统包括两个实体:客户端(被认证者)有一个专门计算密码的密码生成器,认证方有一个密码验证服务器,其交互过程描述如下。

(1) 第一阶段:初始化

如图 3.28 所示,客户端和服务器拥有相同的密码字。客户端向服务器发送一个 S/KEY 初始化消息。服务器端进行应答,应答消息包括一个种子(seed)和一个序列号,即迭代值(iteration),同时还包括算法标识符。

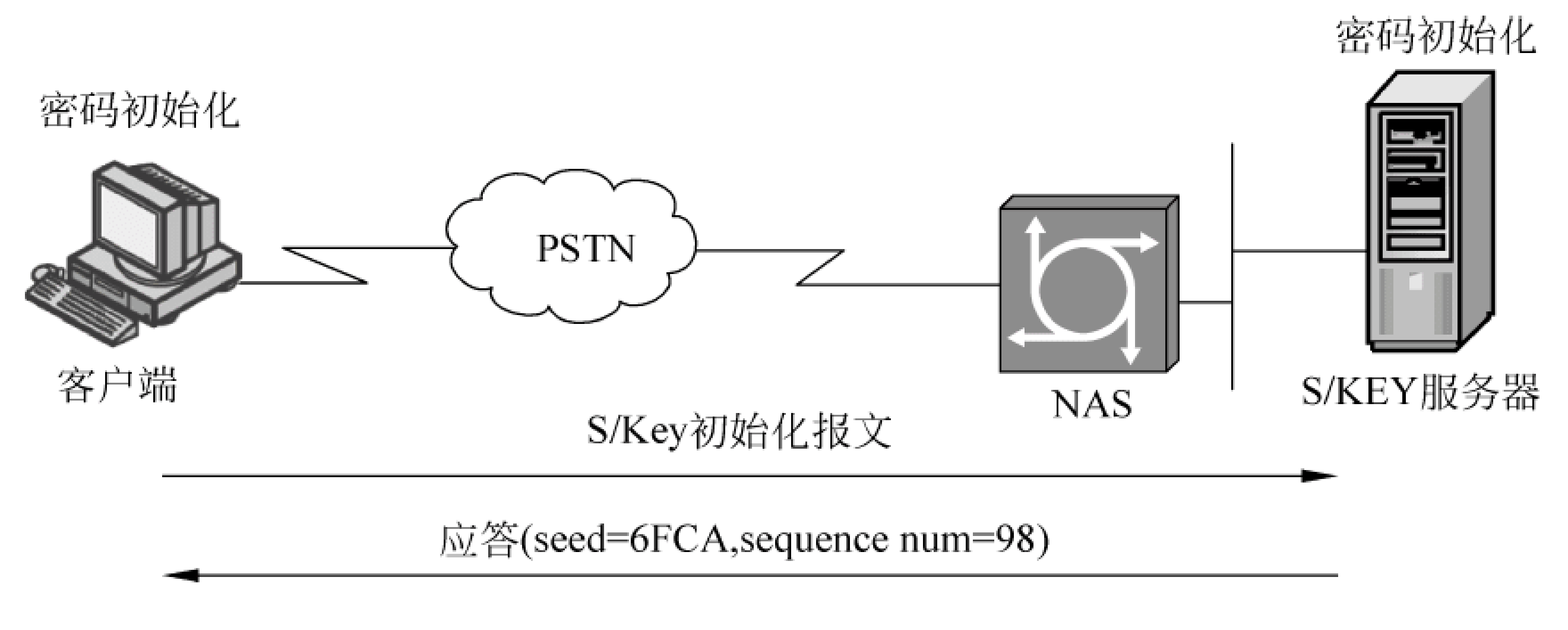


图 3.28 初始化 S/KEY 交换

应答数据包格式为{算法标识符}{序列号}{种子}。其中:

- 算法标识符指定发生器计算密码所用的算法。
- 序列号为一整数 n。
- 种子(seed)是 1～16 位字母与数字组成的字符串。

(2) 第二阶段:S/KEY 密码计算

如图 3.29 所示,发生器生成一次性密码的过程如下。

① 初始化:发生器将通行词与种子相连形成字符串 S,作为算法输入。通行词应该避免使用短密码,以抵御遍历搜索和字典攻击。种子的加入可以增加系统的安全性,使得用户可以在多台机器上使用同一个通行词,通过改变种子的值,用户可以安全地重复使用通

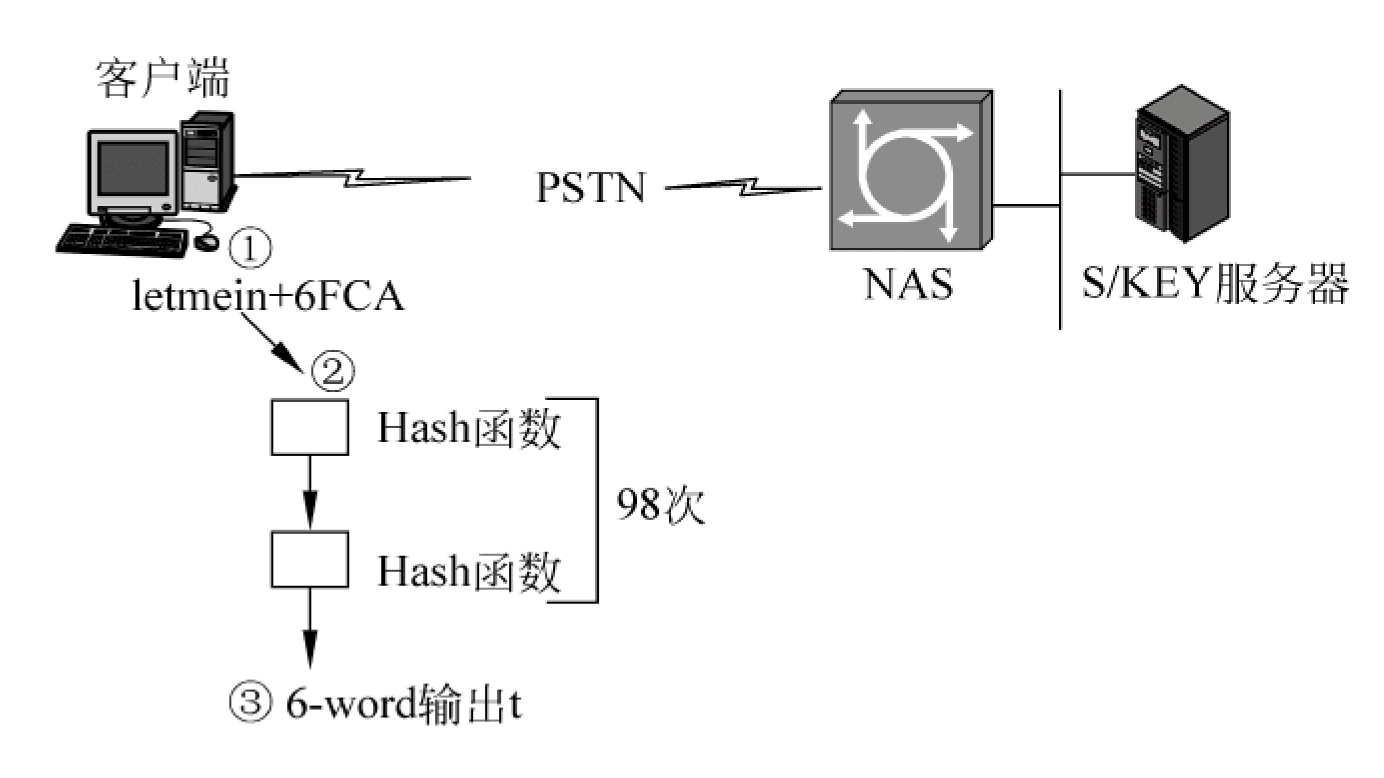


图 3.29 S/KEY 密码计算

行词。

② 密码计算：发生器采用指定的单向函数将 S 经若干遍计算(这里为 n 次)生成一系列密码。第一次使用的密码是 S 经 n 遍单向函数计算得到的结果，即 $hash_n(S)$，下一次使用的密码则是 S 经 $n-1$ 阶单向函数计算所得的值，即 $hash_{n-1}(S)$。以此类推，由于单向函数具有不可逆性，攻击者即使能侦听密码的输出，也不能伪造下一次使用的密码，目前 OTP 系统支持的单向函数有 MD4 和 MD5。

③ 输出：单向函数计算的输出长度为 64 位，这一长度可保证密码系统的安全性。发生器将输出的结果表示为十六进制数列或 6 词序列两种形式返回给服务器。

(3) 第三阶段：S/KEY 密码验证

如图 3.30 所示，服务器将对客户端的密码进行验证。服务器维护一个数据库，保存用户上一次鉴别成功的密码和对应的序列号。对于客户端响应的密码，服务器先将其解码为 64 位的密钥，然后执行一次哈希函数计算，若结果与保存的密码匹配，则鉴别成功，保存新密码；否则，验证失败。

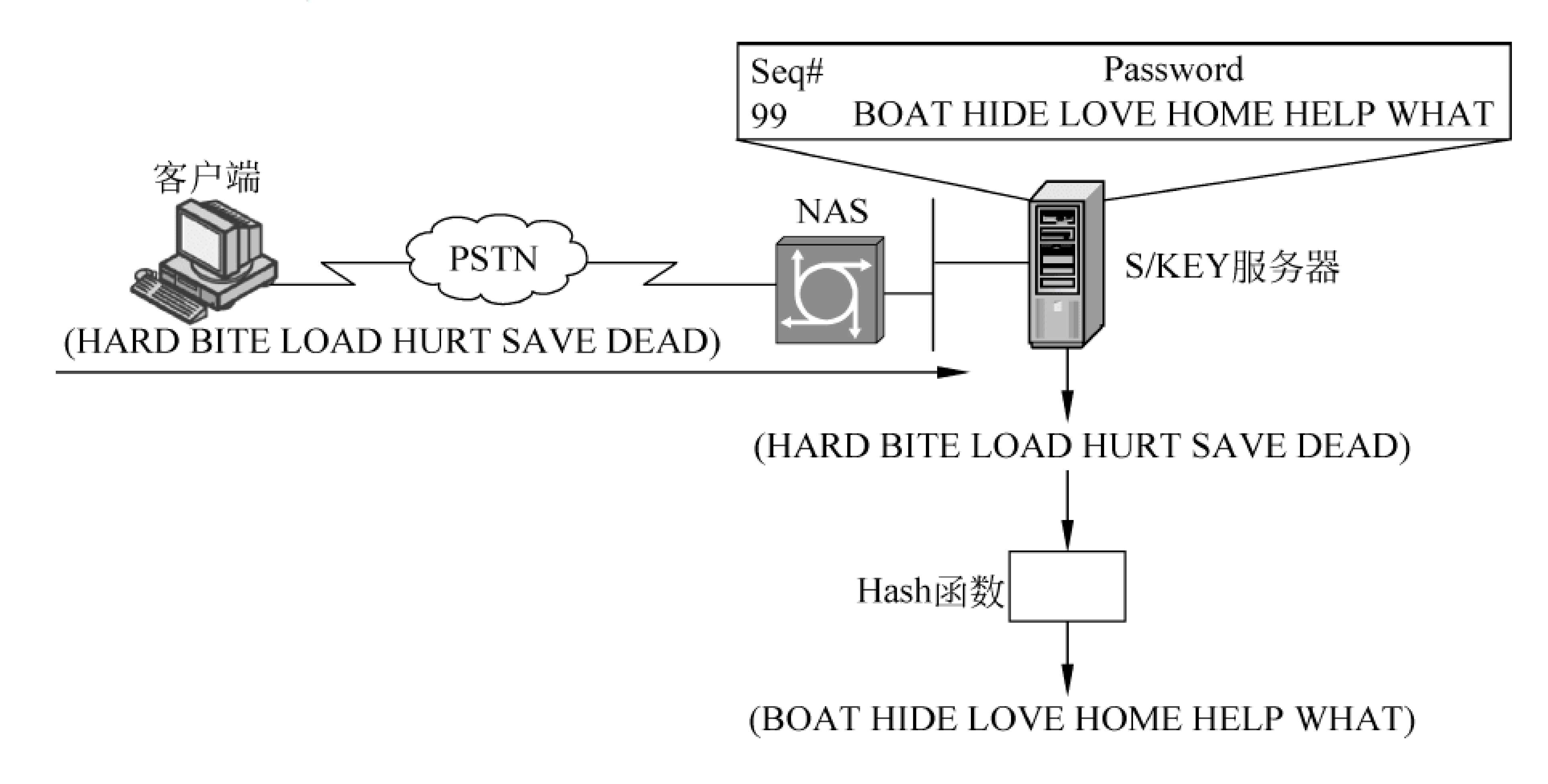


图 3.30 S/KEY 密码验证

本章实验

1. 利用密码破解工具对简单密码进行破解。
2. 利用 Java 和 C++提供的 API 编程实现签名、加密功能。
3. 编程实现验证码。
4. 利用 FREE BSD 的 OPIE 配置和实现一次性密码鉴别。
5. 在 Linux 系统上配置 RADIUS 服务器，实现对用户的认证、授权和记账。

思考题

1. 基于对称密钥的鉴别和基于公钥的鉴别的基础和假设条件分别是什么？从本质上看，这两种鉴别方法有什么联系？

2. 单一身份鉴别和报文鉴别的主要区别是什么？

3. Hash、HMAC 和 MAC 之间的区别是什么？其中哪一种机制既可以进行报文源鉴别，又可以进行数据完整性验证？为什么？

4. Kerberos 鉴别中的许可证(或票据 ticket)的作用是什么？在系统中共有几对密钥，它们分别如何产生？作用分别是什么？

5. 基于挑战/应答的鉴别和一次一密密码 OTP 分别采用什么方法增强密码系统的安全性，两者各有什么优点和缺点？

6. S/KEY 一次性密码鉴别机制中的“种子”有什么作用？

第4章

公钥基础设施

4.1 PKI概述

公钥密码体制可广泛用于加密、身份认证、数字签名等多种安全服务。但是,当提供安全服务的应用系统达到一定规模后,公钥的分发和管理就成为一项十分繁杂的工作。传统公钥密码体制面临的另一个挑战是公钥的可信问题,即如何确定获得的公钥不是假冒的。为了解决公钥的分发、管理以及安全性等问题,目前广泛采用公钥基础设施(public key infrastructure,PKI)技术。PKI是一种使用公钥密码体制实施和提供安全服务的具有普遍性的安全基础设施。PKI采用固定的格式封装公钥,把用户的公钥和用户的其他标识信息(如公钥持有者的名字、序列号和有效期等)捆绑在一起,并使用一个可信的机构在Internet上进行发布。利用PKI基础设施可以有效管理密钥,并提供公钥加密和数字签名服务,保证网上数字信息传输的机密性、真实性、完整性和不可否认性。

目前,PKI实现的通用方法是采用第三方可信机构——认证中心(certificate authority,CA)把用户的公钥封装为数字证书,并利用CA对证书进行签名、管理和分发。这样,公钥可以使用PKI的数字证书获取,而不必使用副本或电子邮件等方式分发给需要使用的用户。

如图4.1所示,数字证书类似于一个证明身份的证件。数字证书的基本内容和实际的证书类似,包括主体名、发布者、公钥和签名等。

- 主体名(subject name)是该证书拥有者的名字,类似于证件上证件持有者的名字。
- 发布者(issuer)是签发该证书的CA,在实用系统中,CA是一个用来产生、保存、管理和发放证书的服务器。
- 公钥(public key)是证书拥有者的公钥。
- 签名(signature)是发布该证书的CA的数字签名,用来确认证书的合法性,即证书

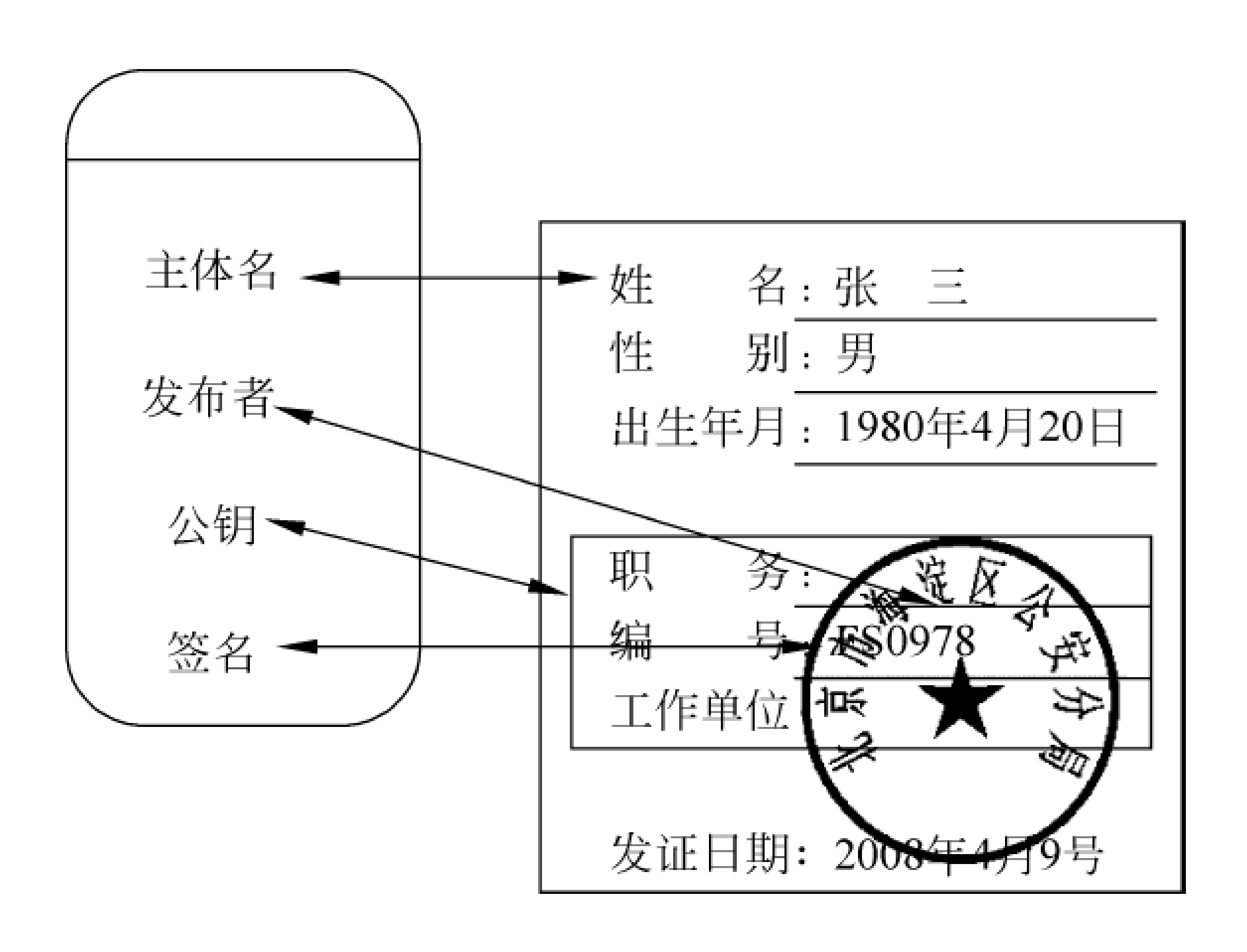


图 4.1　数字证书示意图

的确是由该 CA 签发的。签名类似实际证书中发证机关的签章。证书的使用者在进行证书验证时，首先需要验证 CA 的签名。

X.509 是目前广泛使用的证书格式之一，当然还有其他格式的证书，如 PGP 证书。在 X.509 证书体制中，所有的证书都符合 ITU-T X.509 国际标准。因此从理论上讲，为一种应用创建的证书可以用于任何其他符合 X.509 标准的应用，从而实现不同 PKI 系统证书之间的相互操作。

PKI 中证书的应用和公钥系统中公钥的使用相同。用于加密时，加密前用户 A 需获取通信对等实体 B 的证书，经过证书有效性验证后，从证书中取出 B 的公钥，使用 B 的公钥对数据进行加密，将加密信息发送给 B，B 使用自己的私钥进行解密。用于签名时，用户 A 使用自己的私钥进行签名，签名的验证者 B 首先获取 A 的证书，经过证书有效性验证后，从证书中取出 A 的公钥，使用该公钥对签名进行验证，以确认 A 的身份。

通过自动管理密钥和证书，PKI 可以为用户构建一个安全的网络运行环境，使用户可以在多种应用环境下方便地使用加密和数字签名技术，从而保证数据的机密性、完整性和有效性。一个有效的 PKI 系统必须是安全和透明的，用户在获得加密和数字签名服务时，不需要详细地了解 PKI 是怎样管理证书和密钥的。一个典型、完整、有效的 PKI 应用系统至少应具备以下功能：

- 证书管理。
- 黑名单的发布和管理。
- 密钥的备份、恢复以及自动更新。
- 自动管理历史密钥。
- 支持交叉认证。

4.2 PKI技术发展及应用现状

自20世纪90年代初期以来，作为电子商务信息安全的关键和基础性技术的PKI逐步得到了许多国家的政府和企业的广泛重视，PKI技术由理论研究进入到商业化应用阶段。在这一时期，IETF、ISO等机构陆续颁布了X.509、PKIX、PKCS、S/MIME、SSL、SET、IPSec和LDAP等PKI应用相关标准，RSA、VeriSign、Entrust和Baltimore等企业纷纷推出了自己的PKI产品和服务。一些大的厂商，如Microsoft、Netscape、Novel和Sun等，都开始在自己的网络基础设施产品中增加PKI功能。

PKI技术经过近10年的发展已日趋成熟，许多新技术还在不断涌现，CA之间的信任模型、使用的加解密算法、密钥管理的方案等也在不断变化之中。例如，为了确保电子交易的不可否认性，基于第三方的时间戳（timestamp）服务正在引起人们极大的关注。在PKI的CA网络模型方面，除了传统的层次结构（hierarchy）和对等结构（peer to peer）以外，还出现了桥CA（bridge CA）的概念。PKI的应用也已覆盖了安全电子邮件、虚拟专用网络（virtual private network，VPN）、Web交互安全、电子数据交换、Internet上的信用卡交易等，涉及电子商务、电子政务和电子事务安全等诸多领域，形成了年营业额达数亿美元的大产业，PKI具有非常广阔的市场应用前景。

加拿大、美国、欧盟等国家和地区也相继建立了自己的PKI体系，银行、证券、保险和电信等行业的用户开始接受并使用PKI技术。其他国家和地区也纷纷开始PKI技术的应用，涌现出了众多的认证中心对外提供PKI服务，促进了整个PKI/CA行业的发展。

这些国家和地区开展的PKI服务都有一些共同的特点：政府支持和授权；由政府有关部门实行统一的审核管理；由政府有关部门或民间有关组织制定和发布电子交易法令法规和认证中心认证管理办法；采用有关国际组织发布的技术和操作标准与协议；认证中心的设立需根据有关法令法规严格审批。以上做法为规范、安全、有效地运作PKI/CA奠定了可靠的基础。

韩国是亚洲PKI技术开发较早，且体系相对完善的国家。韩国的认证架构主要分三个等级：最上一级是信息通信部（ministry of information and communication，MIC），中间是由信息通信部设立的国家CA认证中心，最下一级是由信息通信部指定的下级授权认证机构（licensed certificate authority，LCA）。信息通信部还负责相关政策的制定和执行，以及与国外的交叉认证；认证中心则承担根CA的运作和对LCA的评估、支持。与此同时，韩国还在1999年成立了国际CA认证中心。

日本的PKI管理架构也很有特色。首先，它们的应用体系按公众和私人两大领域来划分。其次，把公众领域的应用又进一步进行细分，主要分成商业、政府与公众管理内务、电信邮政三大块。不同的领域适用不同的认证规则，比如在公众领域的三大块当中，就分别采用

商业注册、政府 PKI 以及数字签名法，而私人领域则建立了一种专门的私人 PKI 标准。

一些国家和地区，如美国、欧盟、德国、日本和新加坡等，还相继通过了《电子(数字)签名法》等 PKI 相关法律，在法律上赋予了数字签名与传统手工签名的同等地位，意味着网上证券交易、网上签约和网上政府采购等网上交易行为都可以通过电子签名来完成，极大地推动了 PKI 技术的应用。

随着 PKI 技术的发展以及市场前景的日渐广阔，一些有实力的企业也纷纷投入到这个行业中，成为专业的 PKI 产品与服务提供商，不断推出新的产品与服务，为 PKI/CA 行业的发展推波助澜。在国际上，美国的 VeriSign、加拿大的 Entrust Technologies 和爱尔兰的 Baltimore Technologies 是目前为止全球最大的三家 PKI 产品与服务提供商。

我国的 PKI 应用虽然起步较晚，但 PKI 行业的发展还是十分迅速的。国内的认证中心可分为三大类：行业性认证中心、区域性认证中心和纯商业性认证中心。其中，行业性认证中心主要为特定行业的 PKI 应用提供服务，如中国金融认证中心(China Firarcial Certification Authority，CFCA)、中国电信认证中心(China Telecom Certification Autharity，CTCA)。区域性认证中心主要为当地及周边的用户提供 PKI 服务，如北京数字证书认证中心(Beijing Certificate Authority，BJCA)、上海市电子商务安全证书管理中心有限公司(Shanghai Electrohil Certificate Authority，SHECA)、重庆数字证书认证中心(Chongqing Certificate Authority，CQCA)和广东省电子商务认证中心(Certification and Accreditation Aclministration of the People'Republil of China，CNCA)。除了前两类认证中心以外，国内还有少数纯商业性认证中心，如 iTruschina CA。

4.3 PKI 体系结构——PKIX 模型

PKI 是面向大型开放互连网络应用环境的公开密钥管理机制，它是用以创建、管理、存储、分配和撤销基于非对称加密体制的公钥证书的一组硬件、软件、人员、政策和规程的集合。PKI 系统中，所有用户建立自己的非对称密钥对，并使用全域用户公认可信的第三方——认证中心 CA 来证明其公钥的可靠性。因此，PKI 系统的目标就是向开放网络环境中的用户和应用程序提供可靠的公开密钥管理服务，确保用户能在开放的网络环境中获得真实可靠的公开密钥。

为了实现这一目标，IETF(Internet engineering task force)于 1995 年 10 月成立 PKIX(public key infrastructure X.509)工作组，用来规定以 X.509 作为证书的 PKI 系统的证书概要文件集合和操作模型。PKIX 为不同的应用领域创建了 PKI 模型，并提出了基本 PKI 系统的体系结构。

PKIX 工作组提出的基本 PKI 系统由以下几类实体组成：认证中心、注册中心(registration authority，RA，是一个可选实体)、终端实体 EE(end entity)和证书库

(repository),其基本结构如图 4.2 所示。

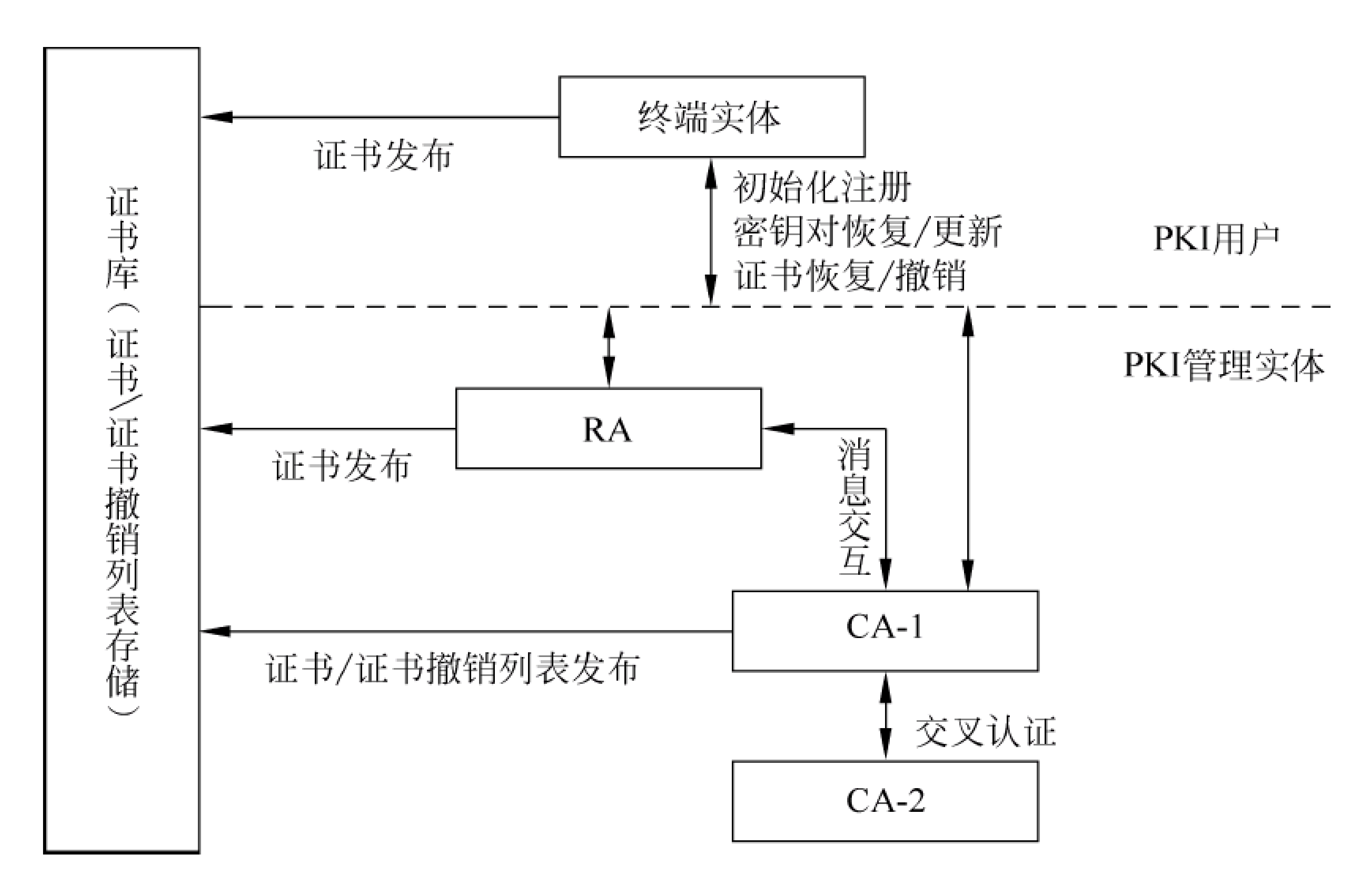


图 4.2 PKIX 模型中定义的 PKI 体系结构

(1) 终端实体

终端实体又称端用户或终端用户,指证书的申请者和持有者,即证书格式中的主体(subject)。在实际系统中,端用户可以是任何一个使用证书的实体,例如一个 WWW 服务器、IE 浏览器、特定的应用系统或程序等,其身份可以是个人、团体和组织等。

为了保证 PKI 系统的安全性,端用户需要构建自身的安全环境,以保证其安全地访问一些重要信息,如端用户的私钥、它信任的 CA 的名字以及该 CA 的公钥等。对于不同的应用系统,这些安全环境可采用不同的方式实现。例如,可以使用经过加密的文件或防篡改的密码令牌(cryptographic tokens)等本地安全存储方式来实现终端实体机密信息的存储。端用户自身安全性的实现方式和程度不属于 PKI 体系的定义范围,由各应用系统根据自身安全需求确定并实现。

如果证书的申请者本身是一个 CA,则端用户即是该 CA,其证书按照 CA 的层次关系(参见 4.5 节)由其上级 CA 进行签发。

(2) 认证中心

认证中心作为权威的、可信赖的第三方机构,负责签发和管理所有参与网上交易的实体所需的数字证书。在实际应用系统中,CA 的功能由一个专门的服务器(如 OpenCA、Windows server CA 或用户自主开发的 CA 服务器)实现,用户自主开发 CA 时,其核心功能可以通过开源的程序组件(如 openssl)实现。

认证中心是电子商务体系中的核心环节,是实现可信电子交易的基础。它通过自身的注册审核体系,检查核实进行证书申请的用户身份和各项相关信息,使证书中标识的信息与网上交易的用户的属性一致。认证中心保证数字证书中列出的公钥持有者的确是其对应公

钥的真实拥有者，同时，认证中心对所签发的数字证书进行数字签名，使得攻击者不能伪造和篡改证书，从而解决了公钥体系中公钥的合法性和可信性的问题。此外，CA 还提供证书发放、证书更新、证书撤销和证书查询等多种证书管理功能。

概括地讲，认证中心的主要功能如下：

① 证书申请处理：接收用户数字证书的申请。

② 证书审批：验证申请者身份，确定是否接受其数字证书申请。

③ 证书签发：为终端实体创建证书并对证书进行签名，然后向申请者颁发数字证书。

④ 证书更新服务：对用户的数字证书进行更新。

⑤ 目录服务：提供用户数字证书的查询。

⑥ 证书撤销服务：接受并处理用户的证书撤销请求。

⑦ 产生和发布证书废止列表(certificate revocation list，CRL)：对于已经被撤销的证书，如果该证书还在有效期内，则 CA 需要将其置于 CRL 中。

⑧ 数字证书的归档：对过期和已废止的数字证书进行归档。

⑨ 密钥管理：负责自身及其下属 CA 的密钥管理。

⑩ 历史数据归档：对各种历史数据进行归档。

(3) 注册中心

对于一个复杂的系统，可以把证书注册和管理等部分功能从 CA 中剥离出来，使用一个独立的注册机构 RA 来实现。将 RA 作为端用户和 CA 之间的一个接口，完成和端用户交互的部分证书管理功能。

注册中心的具体功能实现，不同应用系统有所不同，一般来说，注册中心具有如下功能。

- 自身密钥的管理：包括密钥的更新、保存、使用和销毁等。
- 审核端用户信息：收集端用户信息，并进行证书申请者的身份确认。
- 接收端用户的证书申请。
- 接收并输出证书撤销请求。
- 发布有效证书列表。

对于一个实际的 PKI 应用系统，注册中心是一个可选的实体，它可以独立于 CA 单独存在，也可以是 CA 的一部分。PKI 国际标准和 RFC 文档推荐由独立的 RA 来完成证书的注册管理功能，以增强应用系统的安全性。

如图 4.2 所示，终端实体与 RA 和 CA 之间的消息交互包括：初始化注册(包括证书申请)、密钥对恢复/更新、证书恢复/撤销等申请信息。终端实体向 CA 或 RA 发出以上申请，由 RA 或 CA 处理后将处理结果返回终端实体。RA 和 CA 之间也需要进行各种消息的交互，例如 RA 对终端实体的注册信息进行审核后，将其证书申请转发给 CA。

(4) 证书库

证书库是证书和 CRL 的公共存储，也是 PKI 系统的数据存储中心和发布中心，用于发布通过认证中心认可的 X.509 证书和证书撤销列表。证书库是 PKI 体系的一个重要组成

部分，PKI 大部分组件的管理操作都和证书库密切相关。传统的 PKI 系统一般在文件系统或数据库的基础上实现 PKI 证书库，它通常是一个 X.500 目录。传统的证书库检索方式单一、不易管理，且可移植性、互操作性差，随着轻量级目录访问协议(lightweight directory access protocol，LDAP)标准的日益成熟和 LDAP 目录服务的广泛应用，使用 LDAP 目录服务来设计 PKI 证书库成为目前普遍使用的解决方案。使用 LDAP 对证书进行存储和管理，能够对 PKIX 模型提供更好的支持，并且便于终端用户进行证书的查询和下载。

(5) 交叉认证

在 PKI 的实际应用中，经常遇到不同信任域之间的数字证书交换的问题，需要采用某种机制解决不同 PKI 信任域中各 CA 之间的相互信任问题，这时就需要使用交叉认证技术。

交叉认证使得 CA 可以将其信任范围扩展到其自身的信任域之外，通过交叉认证，不同 PKI 信任域中的 CA 可以互联，使得由某个 PKI 信任域中的 CA 签发的数字证书能够被其他 PKI 信任域中所有的端用户所信任。

假设一个应用系统中涉及 T1 和 T2 两个信任域。T1 域中的可信 CA 为 CA1，T2 域中的可信 CA 为 CA2，如果需要 T1 中的用户信任 T2 中 CA 签发的证书，即实现 CA1 与 CA2 的交叉认证，CA1 需要为 CA2 签发一张交叉认证证书，这张证书中包含 CA2 的公钥(经过 CA1 签名的)，这张证书发布到 T1 域后，T1 中的所有用户就可以信任 CA2 及其子 CA 签发的证书了。

实现交叉认证有多种方法，例如可以将所有其他信任域中的根 CA 嵌入到客户端浏览器中，也可以采用桥 CA 技术解决中小型 PKI 信任域之间交叉认证问题(参见 4.5 节)。

PKIX PKI 体系结构是一个参考模型，给出了 PKI 系统的基本实体及各自的基本功能。在具体实施一个 PKI 系统时，应根据 PKI 应用系统的规模、复杂度以及用户特点，决定系统中实体的种类、数量以及具体的实施和部署方式。例如，对于大型的、提供公共服务的 PKI 系统，可以另外设立一个密钥管理中心(key management center，KMC)为端用户产生非对称密钥对，并把其中的公钥传递给 CA 以封装为证书。这个密钥管理中心可以属于 CA 的一部分，也可以为多个 CA 提供密钥的创建和管理。而对于小型的内部用户使用的自主开发的 CA，则可以在端用户申请证书时，由浏览器产生非对称密钥对，然后将其中的公钥发送给 CA 以创建证书。

此外，PKIX 工作组还发布了三个 RFC 文档，定义了与证书管理相关的协议，包括：证书管理协议(certificate management protocol，CMP)；基于 CMC(cryptographic message syntax)的证书管理消息(certificate management messages over CMS，CMC)；证书请求消息格式(certificate request message format，CRMF)。在这些相关协议中，进一步描述了 PKIX 模型的 PKI 系统中证书请求和管理消息的类型、格式和基本功能。

4.4　X.509 证书

1978 年 Kohnfelder 提出了包含用户名和公钥的签名数据块——证书的概念：数字证书是一段包含用户身份信息、用户公钥信息以及身份验证机构数字签名的数据，它是一个经过证书认证中心(CA)数字签名的包含公开密钥拥有者信息以及公开密钥的文件。它在一个身份和该身份的持有者所拥有的公/私钥对之间建立了一种联系。证书的签名可确保证书在不安全的网络环境中的传输和存储安全，因此，数字证书为开放网络环境中管理公开密钥提供了基本手段。

为了提供公用网络用户目录信息服务，国际电信联盟 ITU 于 1988 年制定了 X.500 系列标准。1988 年，作为 X.500 目录服务的访问授权部分，ITU 提出了 X.509(ISO/IEC 9594-8)证书标准，该标准定义了数字证书的基本内容、格式和证书工作机制。X.509 得到了广泛的应用，是目前普遍使用的证书格式之一。X.509 证书由可信的 CA 创建，并由 CA 或端用户存放于 X.500 目录或 LDAP 目录服务器中。

如图 4.3 所示，X.509 具有不同的版本，各个版本的字段有所不同。目前使用的 X.509v3 在原有版本 X.509v2 和 X.509v1 的基础上进行了扩展。X.509 证书信息由一些标准字段构成，其中每一版本的证书都包含如下信息。

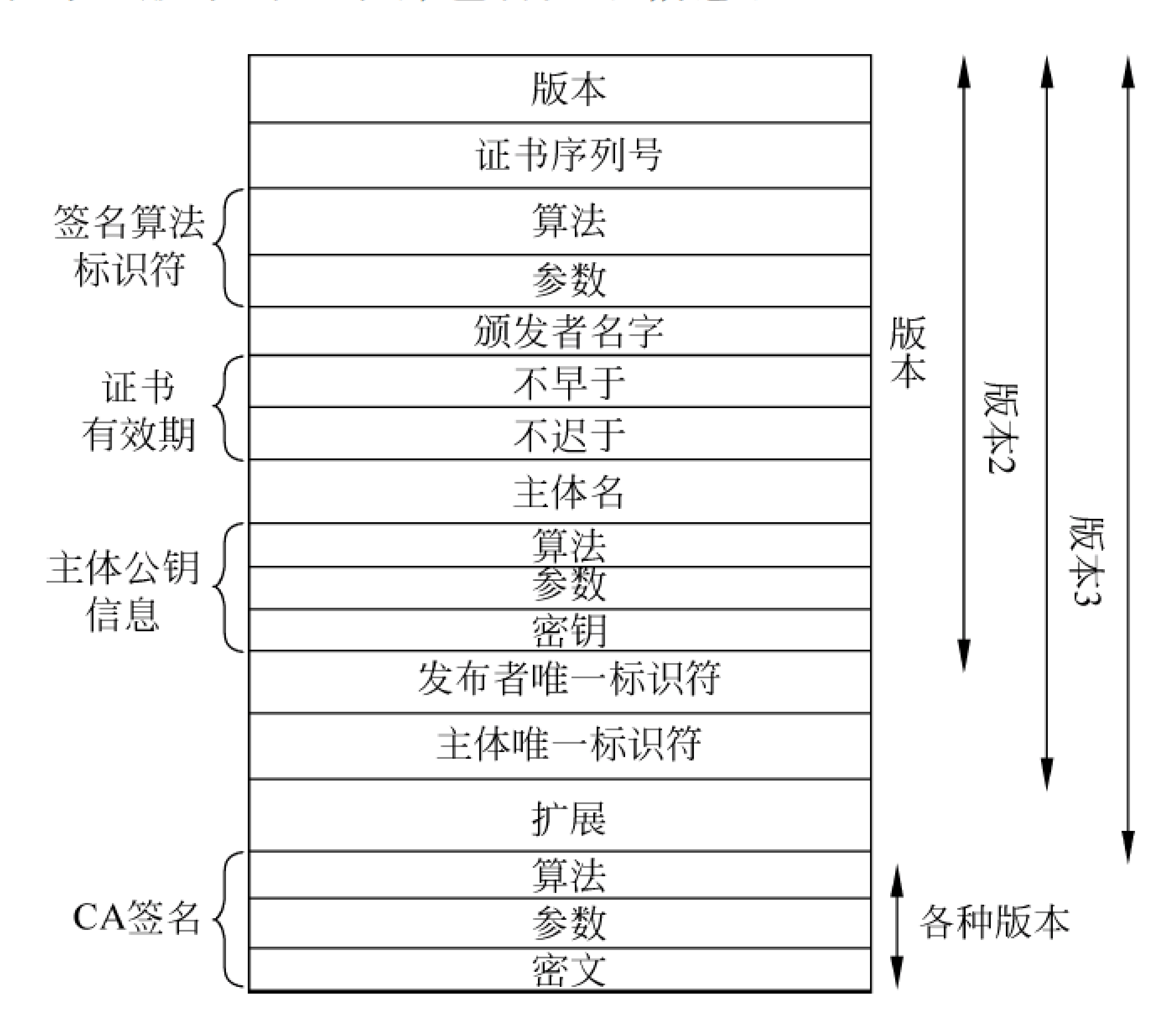


图 4.3　X.509 证书格式

证书的版本号(version)：指明该证书的 X.509 版本号。

证书序列号(certificate serial number)：由 CA 给每个证书分配的唯一编号，当证书被

取消时，该证书序列号放入由 CA 签发的证书作废表或证书黑名单表 CRL 中。

签名算法标识符(signature algorithm identifier)：CA 需要对端用户的证书进行数字签名，以供使用者验证该证书的真实性。该字段用来标识 CA 使用的签名算法及其参数，包括公钥和 Hash 算法。这些算法应该在证书中指明，以便证书的使用者采用同样的算法进行 CA 签名验证。

证书颁发者名字(issuer name)：指明颁发该证书的 CA 的可识别名(distinguished name，DN)。

证书有效期(period of validity)：指明使用该证书的有效期限，包括证书生效时间和证书失效时间。

证书主体名(subject name)：证书拥有者的可识别名，这个字段必须是非空的，除非在证书扩展中使用了主体别名。

证书主体的公钥信息(subject's public key information)：包括证书持有者的公钥算法、参数和公钥等。证书的使用者将这些信息中指明的算法、参数和公钥进行加密或签名验证等安全操作。

证书颁发者标识符(issuer unique identifier)：证书颁发者的唯一标识符，仅在版本 2 和版本 3 中有要求，属于可选项。

证书主体标识符(subject unique identifier)：证书拥有者的唯一标识符，仅在版本 2 和版本 3 中有要求，属于可选项。

证书颁发者的签名(signature)：为证书颁发者使用其私钥生成的签名，以确保这个证书是由可信机构发布的，并且在发放之后没有被篡改过。证书的使用者将使用签名算法标识符字段中指定的签名算法及其参数验证该签名的真实性、消息的完整性。

扩展(extensions)：可选的标准和专用的扩展(在版本 3 中使用)，它们包括如下内容。

- 密钥标识符：证书所含密钥的唯一标识符，用来区分同一证书拥有者的多对密钥。
- 密钥使用：一个比特串，指明(限定)证书的公钥可以完成的功能或服务，如证书签名、数据加密等。
- 扩展密钥使用：由一个或多个对象标识符(object identifiers，OIDs)组成，可以说明证书密钥的特殊用途。有 Internet 策略限定。
- CRL 分布点：指明 CRL 的分布地点。
- 私钥的使用期：指明证书中与公钥相联系的私钥的使用期限，它也由 Not Before 和 Not After 组成。若此项不存在时，公私钥的使用期是一样的。
- 证书策略：由对象标识符和限定符组成，这些对象标识符说明证书的颁发与使用策略有关。
- 策略映射：表明两个 CA 域之间的一个或多个策略对象标识符的等价关系，仅在 CA 证书里存在。
- 主体别名：指出证书拥有者的别名，如电子邮件地址、IP 地址等，别名是和 DN 绑定

在一起的。

- 颁发者别名：指出证书颁发者的别名，如电子邮件地址、IP 地址等，但颁发者的 DN 必须出现在证书的颁发者字段。
- 主体目录属性：指出证书拥有者的一系列属性。可以使用这一项来传递访问控制信息。

为了使 X.509 证书适用于大规模的网络环境中的分发和管理，X.509v3 在其证书格式中增加了扩展字段，提供更多的灵活性和可扩展性以满足各种应用系统不同类型的安全需求。X.509v3 证书包括如下新特征。

（1）多算法支持

X.509 证书是算法独立的，CA 可以根据需要选择证书的签名和消息摘要算法以及端实体所拥有密钥对的类型。在 X.509v3 中，新的算法只需定义算法标识符，公钥格式和密钥使用途径扩展项就可以应用到 X.509 证书系统中，从而保证算法和证书的独立性，支持新算法的使用。目前 IEFT PKIX 工作组支持以下算法。

① 消息摘要算法：MD2、MD5、SHA－1。

② 证书签名算法：RSA、DSA。

③ 密钥持有者的密钥对类型：RSA 密钥、DSA 签名密钥、D—H 密钥交换密钥、KEA 密钥和 ECDSA 密钥。

（2）多种命名机制支持

X.509v1 中使用了 X.500 名字机制来标识确认者和持证者。然而 X.500 的全局目录并未被广泛接受，因此有必要引入其他命名机制。为此，X.509v3 定义了 IssueAltName 和 SubjectAltName 两个扩展项。这两个扩展字段目前支持的命名机制包括 E-mail 地址（RFC822 name）、IP 地址、域名（DNS name）和 URL。一个证书中可以同时使用多种命名方式，但由于 CA 不能确定验证者将使用哪个名字，因此必须保证持证者的每一个名字都是经过确认的。

（3）限制证书（公钥）的用途

X.509v3 证书通过定义 KeyUsage 和 ExtKeyUsage 两个扩展字段，使 CA 能够限制它所发布的证书的用途。目前支持的用途包括签名、不可否认性、密钥加密、数据加密、密钥协商、证书签发和 CRL 签发。其他的密钥用途可以使用 ExtKeyUsage 字段进行扩展。

（4）定义证书遵循的策略

每个 CA 都定义了一定的安全策略，以规范它的证书操作过程。证书实践声明（CPS）集中描述了这些策略，包括该 CA 的命名空间、身份验证、撤销机制、法律责任和收费等。

每个证书都是在一定的安全策略指导下签发的，X.509v3 的 PolicyInfomation 扩展字段可以指明该证书所遵循的 CPS 的位置。

（5）证书链处理

由于一个公钥用户拥有的可信证书管理中心数量有限，要与大量不同管理域的用户建

立安全通信，需要在 CA 之间建立信任关系。一般对于证书链的处理需要考虑与每个证书相关联的信任关系。X.509v3 定义了以下扩展字段用于控制信任关系的传递。

① Basic Constraints：定义持证者是否是 CA 以及允许起始于该证书的证书链的深度。

② Name Constraints：若持证者是 CA，可用这个字段限制该 CA 签发的证书必须在特定的命名空间内。

③ Policy Constraints：通过在 v3 证书的 PolicyConstraints 扩展字段中指明特定的策略号 ID，认证中心可以要求自己所信任的其他 CA 也遵循一定的安全策略。

4.5 PKI 信任模型

信任模型(trust model)是构建和运作 PKI 所必需的环节。选择正确的信任模型以及与它相应的安全级别是非常重要的，同时也是部署 PKI 所要做的基本决策之一。

PKI 信任模型主要解决如下问题：

- 一个 PKI 用户能够信任的证书是怎样被确定的？
- 这种信任是怎样被建立的？
- 在一定的环境下，这种信任如何被控制？

为了进一步说明信任模型，首先给出几个重要的概念。

- 信任　ITU-T X.509 规范中给出信任的定义是“如果一个实体假定另一个实体会严格并准确地按照它所期望的那样行动，那么它就信任该实体”。其中的实体是指在网络或者分布式环境中具有独立决策和行动能力的终端、服务器或者智能代理等。在 PKI 体系中，可以把信任的概念定义为：如果一个用户假设能够建立并维持一个准确的对公开密钥属性的绑定，并且能够明确 CA 所颁发证书的实体的真实身份，则该用户信任该 CA。这样，PKI 中的信任体系也就可以建立起来了。
- 信任域　信任域是公共控制下或服从一组公共策略的系统集，简单来说就是信任的范围。识别信任域及其边界对于构建 PKI 很重要。一个信任域中的用户如果需要使用另一信任域中签发的证书则需要进行交叉认证，通常比使用同一个信任域内签发的证书复杂得多。信任域可以按照组织结构和地理界限来划分。能否建立一个可确定本地信任模型的广泛策略的信任域，对于 PKI 部署的成败具有重要影响。
- 信任锚　在信任模型中，当可以确定一个实体身份或者有一个足够可信的身份签发者证明该实体的身份时，另一个实体就能做出对它信任的决定，这个可信的身份签发者就称为信任锚(trust anchor)。简单地讲，信任锚就是 PKI 信任模型中信任的起点。
- 信任关系　证书的用户要找到一条从证书颁发者到信任锚的路径可能需要建立一

系列的信任关系。在公钥基础设施中，当两个认证机构中的一方给另一方的公开密钥颁发证书时，两者之间就建立了信任关系。在一个实体需要确认另一个实体身份时，它首先需要确定信任锚，再由信任锚找出一条到达待确认实体的各个证书，这些证书组成的路径称为信任路径，通过信任路径可以进行信任关系的传递。信任关系可以是双向的也可以是单向的。

PKI 的常用的信任模型包括单信任模型、严格层次结构模型（strict hierarchy of certification authorities model）、分布式信任结构模型（distributed trust architecture model）、桥 CA 信任模型（bridge CA model）、Web 信任模型（web model）和以用户为中心的信任模型（user centric trust model）。

1. 单信任模型

单信任模型是最基本的信任模型，也是在小型企业环境内部比较实用的一种模型。在这种模型中，整个 PKI 体系只有一个 CA，它为整个体系中的所有终端用户签发和管理证书。PKI 体系下的所有终端用户都信任这个 CA。每个证书路径都起始于该 CA 的公开密钥，该 CA 的公开密钥成为 PKI 体系中所有用户唯一的信任锚。

如图 4.4 所示，这种信任模型的优点是结构简单、容易实现、易于管理，只需要建立一个 CA，所有的终端用户都能实现相互认证。同时，证书路径就是从 CA 直接到各个终端用户，证书策略十分简单。它的缺点也是显而易见的，如所有用户只能从一个组织获取证书，不易扩展信任域，难以实现交叉认证；无法支持大量的或者不同的群体用户，终端用户的群体越大，CA 的负担就越重。

因此，该信任模型只适合用户较少且集中的单一信任域的系统。

2. 严格层次结构模型

如图 4.5 所示，对于一个运行 CA 的较大的认证机构而言，签发证书的工作不能仅由一个 CA 来完成，可以建立一个 CA 层次结构模型，该模型是一个以主从关系建立的分级结构。

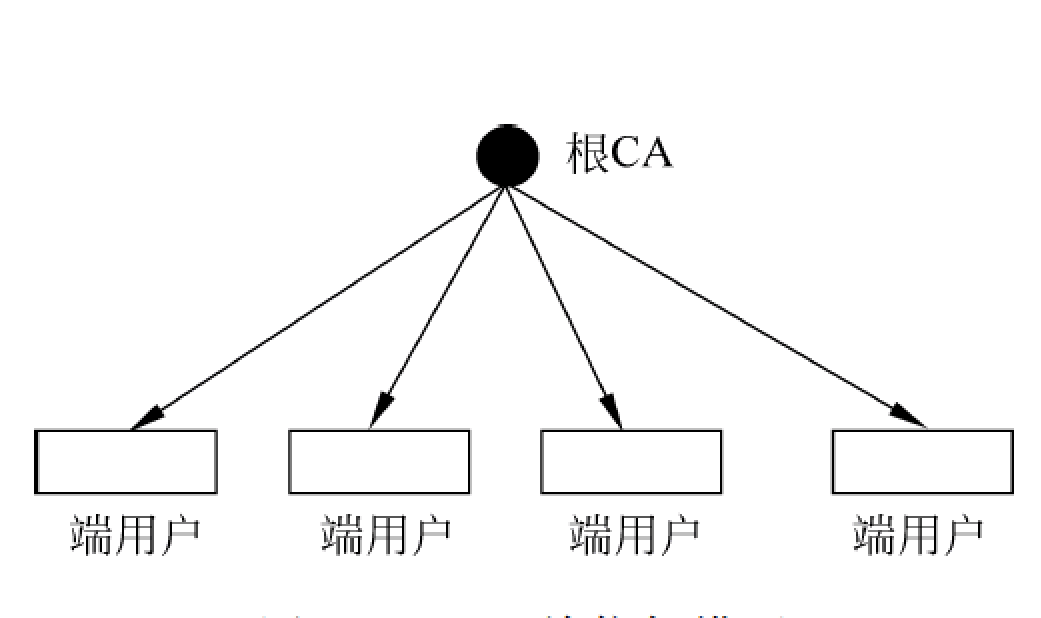


图 4.4　CA 单信任模型

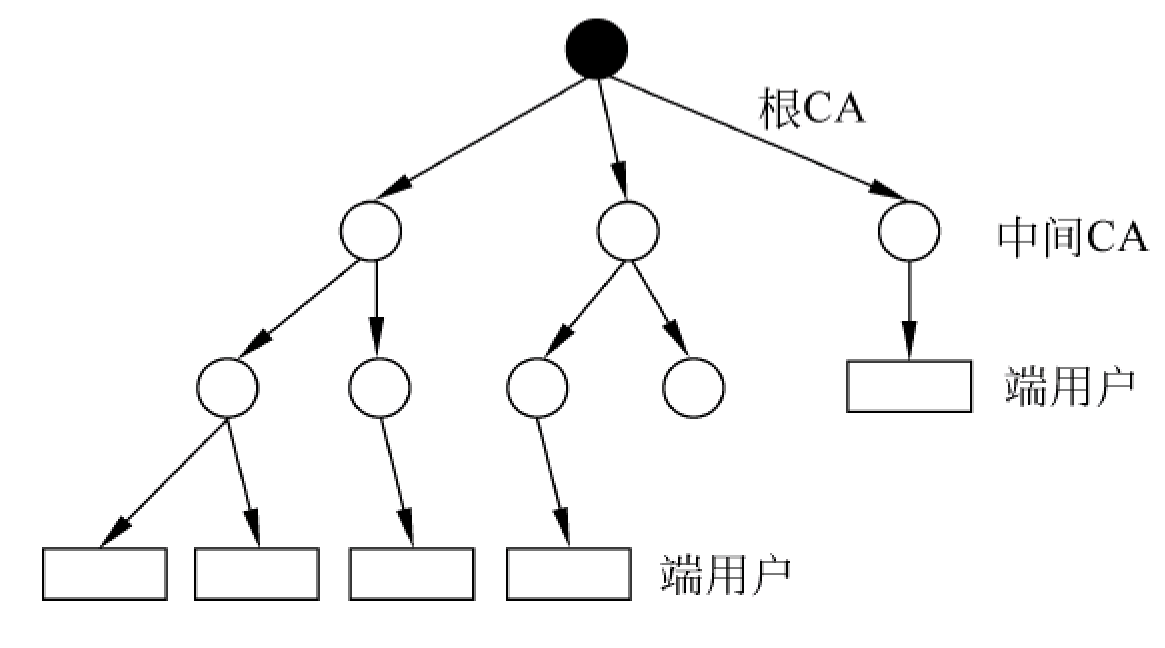


图 4.5　CA 严格层次结构模型

CA 的严格层次结构可以被描绘为一个树形结构，根代表一个对整个 PKI 系统的所有实体都有特别意义的 CA——通常叫做根 CA(root CA)，它充当信任的根或信任锚——也就是认证的起点或终点。在根 CA 的下面是零层或多层中间 CA(inter mediate CA)，也称作子 CA(subordinate CA)，它们从属于根 CA。子 CA 用中间节点表示，从中间节点再伸出分支。与非 CA 的 PKI 实体相对应的叶节点即为端用户实体。在这个模型中，层次结构中的所有实体都信任唯一的根 CA。这个层次结构按如下规则建立：

① 根 CA 具有一个自签名的证书。

② 根 CA 认证(生成和签发证书)与其直接相连的 CA。

③ 每个 CA 都认证零个或多个与其直接相连的下一级 CA(在一些 CA 严格层次结构中，上层的 CA 既可以认证其他 CA 也可以认证终端实体。虽然在现有的 PKI 标准中并没有排除这一点，但是在文献中层次结构往往都是假设一个给定的 CA 要么认证终端实体要么认证其他 CA，但不能两者都认证)。

④ 倒数第二层实体(即最底层的 CA)认证终端用户。

⑤ 对于终端用户，它需要信任根 CA，对于中间的 CA 则可以不必关心(对终端用户透明)。

在 CA 严格层次结构中，每个实体(包括中间 CA 和终端实体)都必须拥有根 CA 的公钥，该公钥的安装是在这个模型中为随后进行的所有通信进行证书处理的基础。因此，它必须通过一种安全的方式来完成。例如，一个实体可以通过物理途径如信件或电话来取得这个密钥；也可以选择通过电子方式取得该密钥，然后再通过其他机制来确认它，如将密钥的散列结果用信件发送、公布在报纸上或者通过电话告之。

在证书验证时，沿着层次关系的树形结构往上找，可以构成一条证书链，直到根证书。而证书中签名的验证过程是沿相反的方向，从根证书开始，依次往下验证每一个证书中的签名，并信任对应的 CA 的公钥，直到最后验证终端用户证书中最底层的 CA 的签名，并最后信任终端用户的公钥。

下面举例说明在 CA 的严格层次结构模型中进行证书验证的过程。一个持有根 CA 公钥的终端实体 A 可以通过下述方法检验另一个终端实体 B 的证书。假设 B 的证书是由 CA2 签发的，而 CA2 的证书是由 CA1 签发的，CA1 的证书又是由根 CA 签发的。A 由于拥有根 CA 的公钥 K_R，因此它能够验证 CA1 的公钥 K_1，并且它可以提取出可信的 CA1 的公钥。然后，这个公钥 K_1 可以被用作验证 CA2 的公钥，类似地就可以得到 CA2 的可信公钥 K_2。公钥 K_2 能够被用来验证 B 的证书，从而得到 B 的可信公钥 K_B。A 现在就可以根据需要使用密钥 K_B，如对发给 B 的消息进行加密，或用来验证 B 的数字签名，从而实现 A 和 B 之间的安全通信。

3. 分布式信任模型

如图 4.6 所示，分布式信任模型也称为网状信任模型。在该模型中，信任锚的选取不是唯一的，存在多个根 CA，CA 之间存在交叉认证。如果任意两个 CA 之间都存在着交叉认

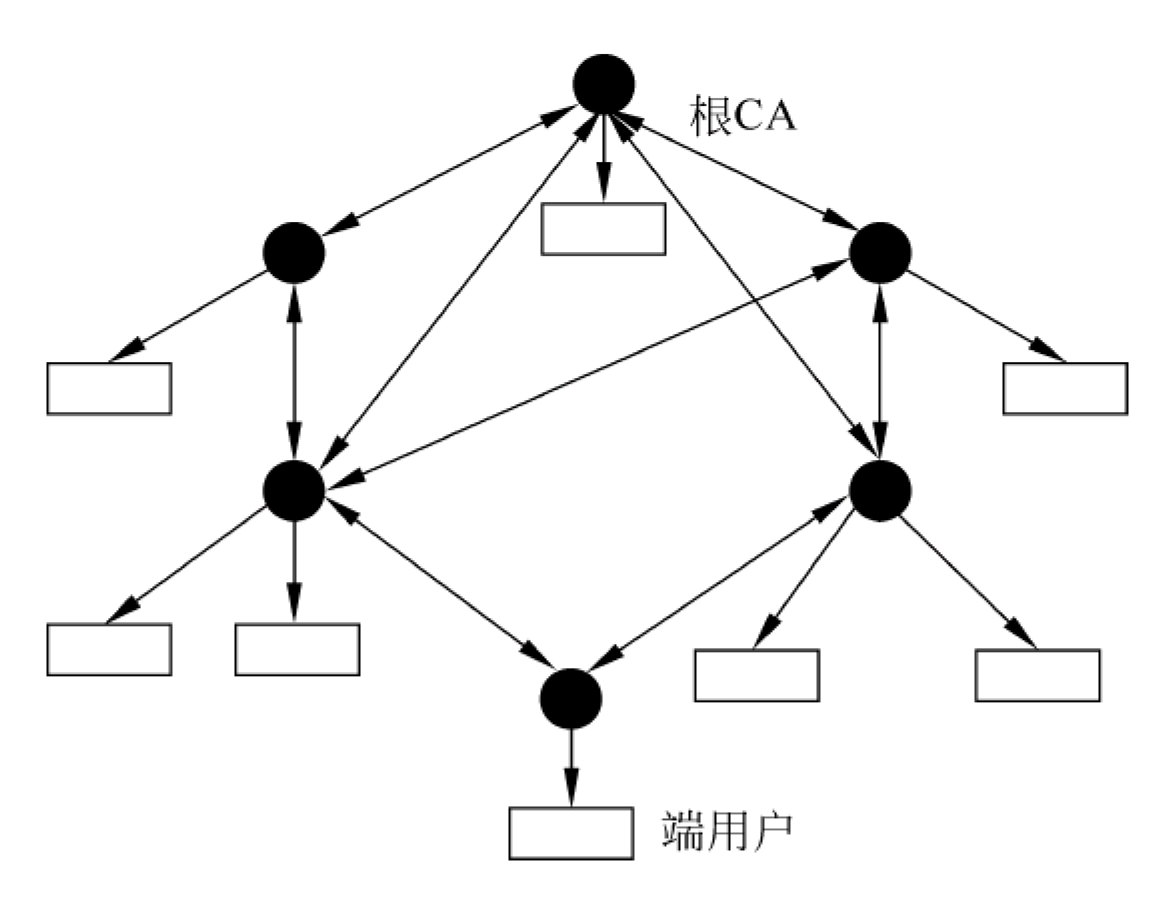


图 4.6 CA 分布式信任模型

证,则这种模型就称为严格网状信任模型。与在 PKI 体系中的所有实体都信任唯一根 CA 的严格层次结构模型不同,网状信任模型把信任分散到两个或多个 CA 上。

由于存在多个信任锚,分布式信任模型中单个 CA 安全性的削弱不会影响到整个 PKI。并且,增加新的认证域比较容易,只需新的根 CA 在网络中至少向另一个 CA 发放证书,用户不需要改变信任锚。由于信任关系可以传递,因此可以减少颁发证书的个数,使证书管理更加简单容易,这种模型能够很好地适应企业之间信任关系的建立。

分布式信任结构模型中,信任关系的传递使得从终端用户证书到信任锚建立证书的路径不确定,因此信任路径的发现比较困难。

4. 桥 CA 信任模型

在桥 CA 信任模型中,采用一个专门的称为桥 CA 的认证机构来连接不同的 PKI 体系,以建立交叉认证。不同于网状模型中的 CA,桥 CA 与不同的信任域建立对等的信任关系,允许用户保持原有的信任锚。这些关系被结合起来形成信任桥,使得来自不同信任域的用户通过指定信任级别的桥 CA 相互作用。

如图 4.7 所示,桥 CA 不是一个树状结构的 CA,也不像网状 CA,它不直接向用户颁发证书。它不像根 CA 一样成为一个信任锚,它只是一个单独的 CA。它与不同的信任域之间建立对等的信任关系,任何结构类型的 PKI 都可以通过桥 CA 连接在一起,实现彼此间的信任,每一个单独的信任域都可以通过桥 CA 扩展到整个 PKI 体系中。

5. Web 信任模型

该模型建构在浏览器的基础上,如图 4.8 所示,浏览器厂商在浏览器中内置了多个根 CA,每个根 CA 相互间是平行的,浏览器用户信任这些根 CA 并把它们作为自己的信任锚。浏览器上预装的这些公钥确定了一组浏览器用户最初信任的根 CA,而这些根 CA 又可构成

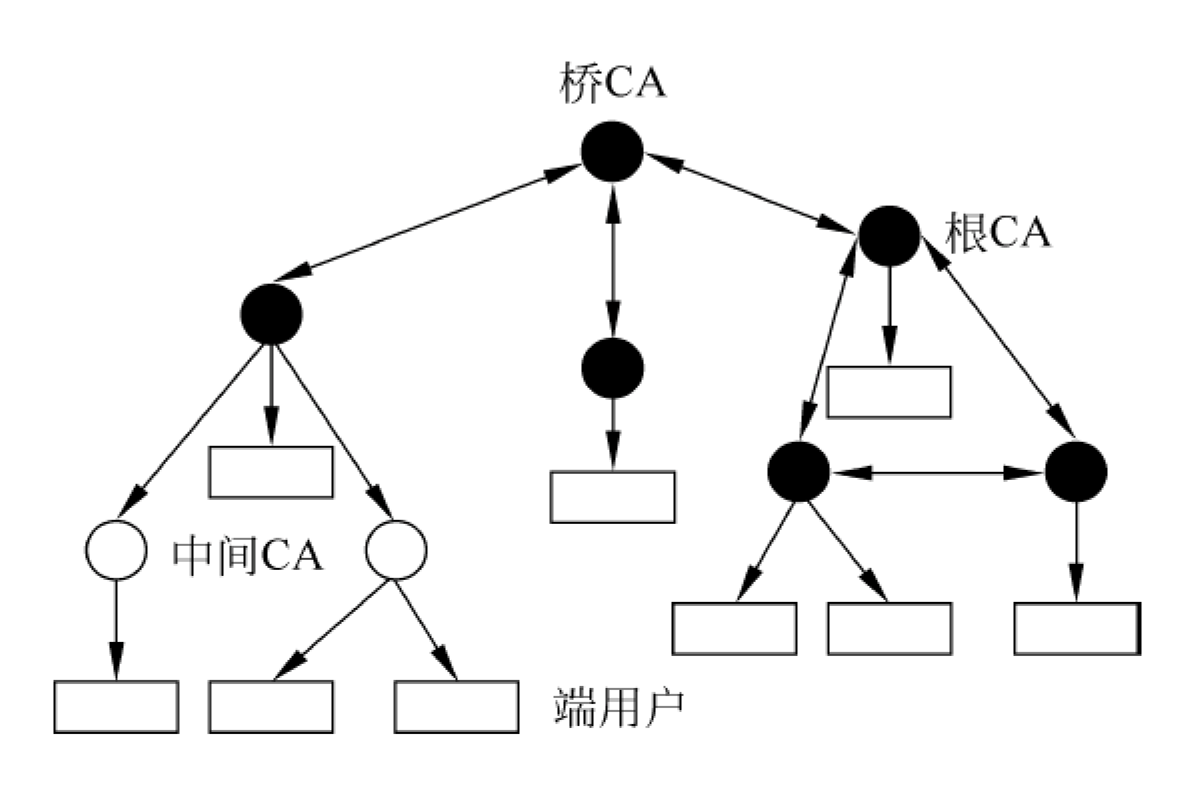


图 4.7　桥 CA 信任模型

图 4.9 所示的信任体系。在该模型中,浏览器厂商就成为了事实上的隐含的根 CA。这种模型表面上与分布式信任模型颇为相似,实际上,它更接近严格分级模型。



图 4.8　Web 信任模型中受信任的根 CA

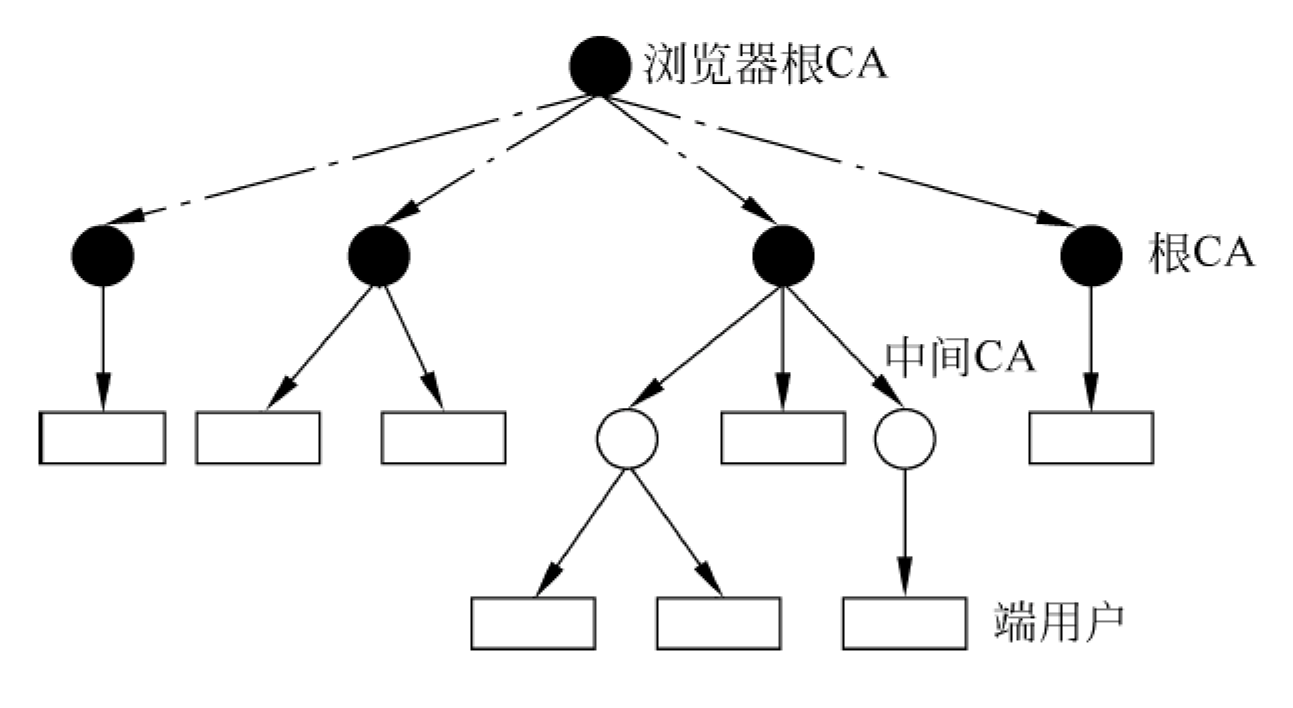


图 4.9　Web 信任模型

Web 信任模型在方便性和简单互操作性方面有明显的优势,但是也存在许多安全隐患。例如,因为浏览器的用户自动地信任预安装的所有公钥,所以如果这些根 CA 中有一个

是不合法的，PKI 体系的安全性将被完全破坏。

另外一个潜在的安全隐患是没有实用的机制来撤销嵌入到浏览器中的根密钥。如果发现一个根密钥是不合法的，或者与根的公钥相应的私钥被泄密了，要使全世界数百万个浏览器都自动地废止该密钥的使用是不可能的，这是因为无法保证通报的报文能到达所有的浏览器，而且即使报文到达了浏览器，浏览器也没有处理该报文的功能。因此，从浏览器中去除坏密钥需要全世界的每个用户都同时采取明确的动作；否则，一些用户将是安全的而其他用户仍处于危险之中。但是这样一个全世界范围内的同时动作是不可能实现的。

最后，该模型还缺少有效的方法在 CA 和用户之间建立合法协议，该协议的目的是使 CA 和用户共同承担责任。

6. 以用户为中心的信任模型

在以用户为中心的信任模型中，每个用户自己决定信任哪些证书。安全软件(pretty good privacy，PGP)最能说明以用户为中心的信任模型。例如，当 Alice 收到一个声称为 Bob 的证书时，她发现这个证书是由她不认识的 David 签署的，而 David 的证书则是由她信任的 Catherine 签署的。在这种情况下，Alice 可以决定信任 Bob 的密钥(即信任从 Catherine 到 David 再到 Bob 的密钥链)，也可以决定不信任 Bob 的密钥(不信任这种密钥链)。

以用户为中心的信任模型需依赖于用户自身的行为和决策能力，因此这种信任模型在技术水平较高和利害关系高度一致的群体中是可行的，但是在一般的群体(它的许多用户缺乏安全意识及 PKI 的概念)中是不现实的。而且，这种模型一般不适合用在贸易、金融或政府环境中，因为在这些环境下，通常希望或需要对用户的信任实行某种控制，显然这样的信任策略在以用户为中心的模型中是不可能实现的。

4.6 密钥和证书的生命周期

4.6.1 密钥/证书生命周期管理

如图 4.10 所示，PKI 体系中，密钥/证书生命周期管理过程可分为初始化、颁发和取消三个阶段，它是对密钥和证书生命周期进行管理的过程，该过程的主体是 CA 或 RA，客体是证书和密钥，管理过程涉及 PKI 体系的各个实体。

(1) 初始化阶段

在端用户实体能够使用 PKI 支持的服务之前，它们必须初始化以进入 PKI，证书管理的初始化阶段由以下几个步骤组成。

① 终端实体注册：终端实体向 CA/RA 提交注册信息并发布证书请求。

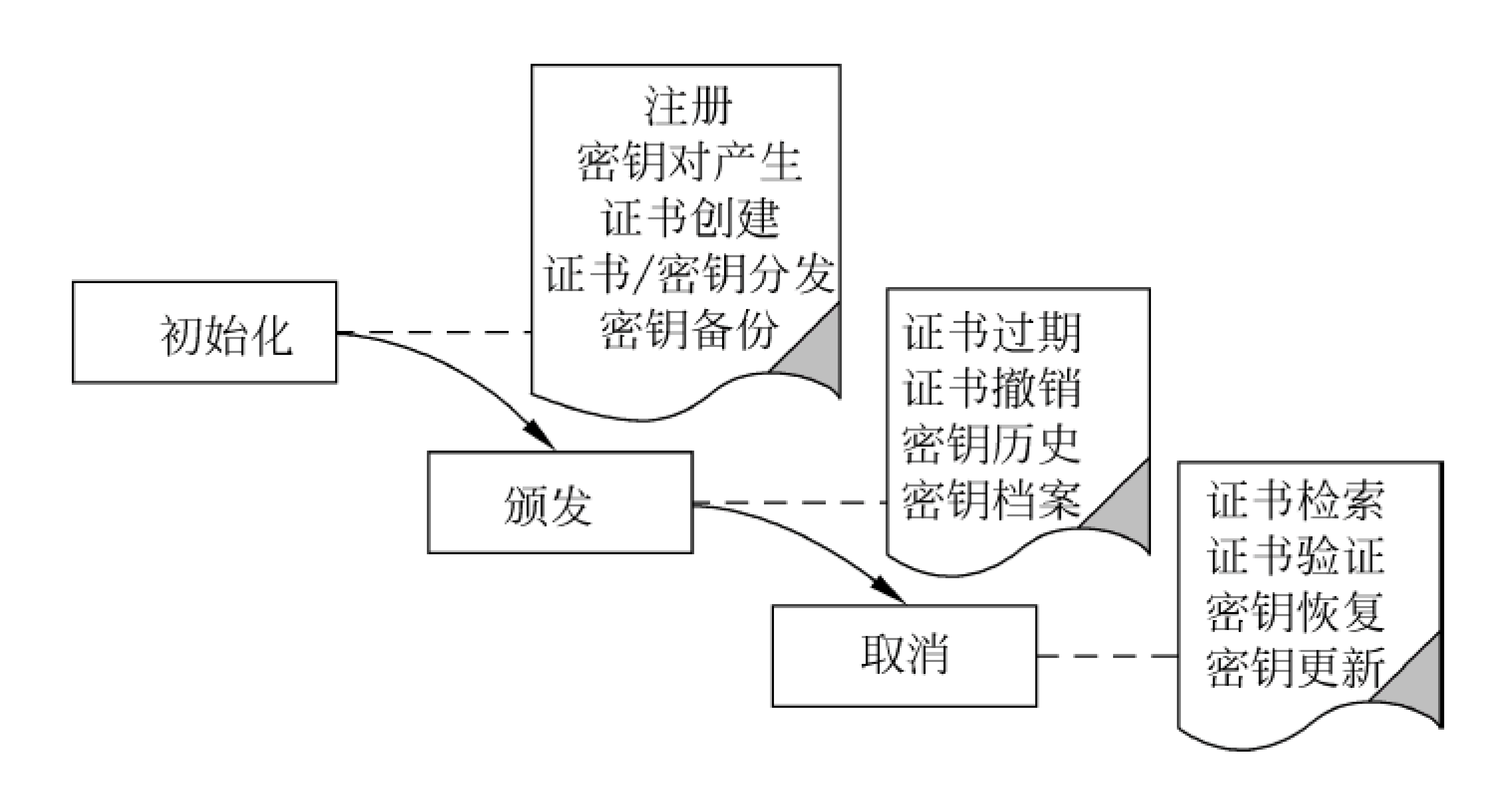


图 4.10 密钥/证书生命周期管理过程

② 密钥对产生：由终端实体或 CA 的密钥管理软件产生公钥/私钥对。

③ 证书创建：CA 为终端实体创建证书并进行签名。

④ 证书/密钥分发：CA 签发证书后，将其发布到指定的资源中，如 CA 数据库或目录服务器中。

⑤ 密钥备份：CA 的密钥管理中心对用户私钥进行备份，以便进行密钥恢复。

其中，一个典型的端用户实体注册过程如图 4.11 所示。端用户向 RA 发起注册表格请求，RA 进行应答后，端用户向 RA 提交注册表格，RA 将该表格提交 CA 进行审核，CA 审核通过后建立注册信息，把注册结果返回端用户。随后，开始证书申请过程。

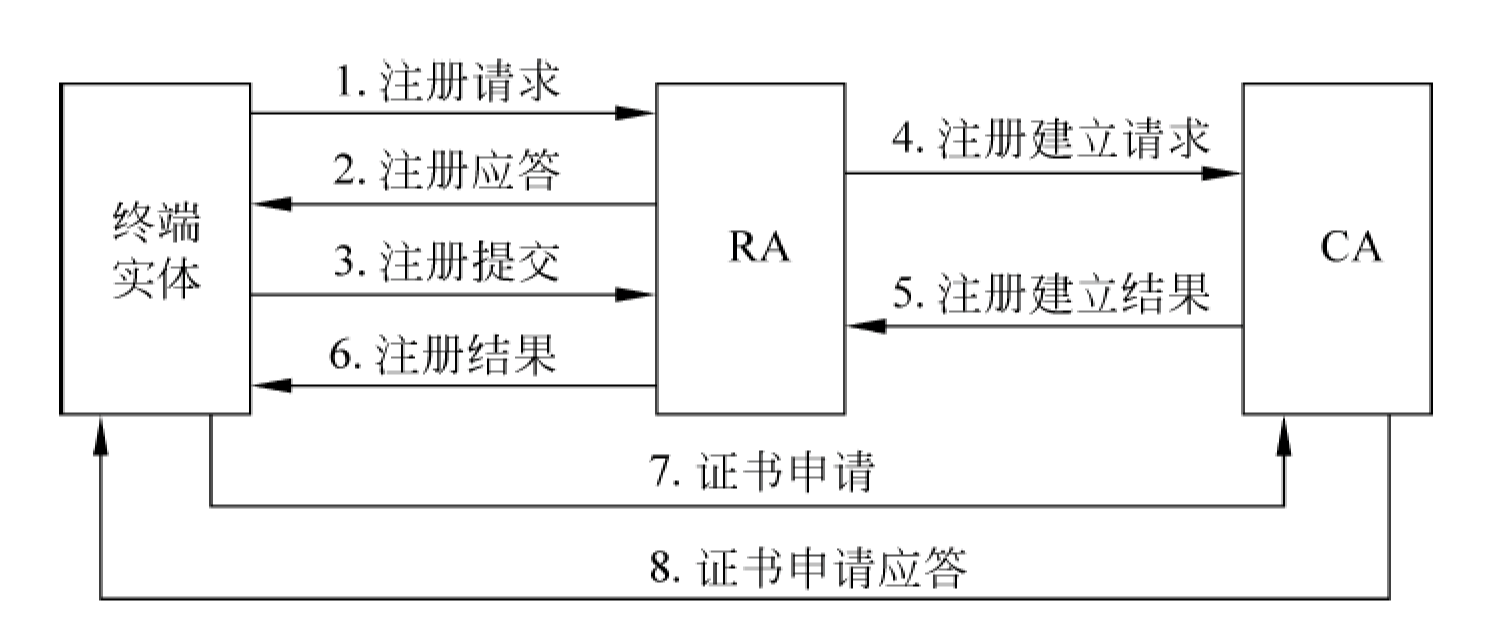


图 4.11 终端实体的初始化

（2）颁发阶段

一旦数字证书已经产生并适当地分发，密钥/证书生命周期管理的颁发阶段即开始。这个阶段包括如下内容。

① 证书检索：进行远程资料库的证书检索。

② 证书验证：确定一个证书的有效性（包括证书路径的验证）。

③ 密钥恢复：当不能正常访问密钥时，从 CA 或可信第三方处恢复密钥。

④ 密钥更新：当一个合法的密钥对过期时，进行新的公/私钥的自动产生和相应证书的颁发。

(3) 取消阶段

密钥/证书生命周期管理以取消阶段来结束。此阶段包括如下内容。

① 证书过期：证书到达有效期后处于过期状态，是证书生命周期的自然结束。

② 证书撤销：在证书有效期内，宣布一个合法证书(及其相关私钥)不再有效。

③ 密钥历史：维持一个有关密钥资料的记录(一般是关于终端实体的)，以便那些使用过期的密钥所加密的数据能够被解密。

④ 密钥档案：出于对密钥历史恢复、审计和解决争议等所进行的密钥资料的安全第三方储存。

4.6.2　密钥生命周期

如图 4.12 所示，对应于密钥/证书生命周期管理这三个大的阶段，密钥的生命周期包括如下阶段：

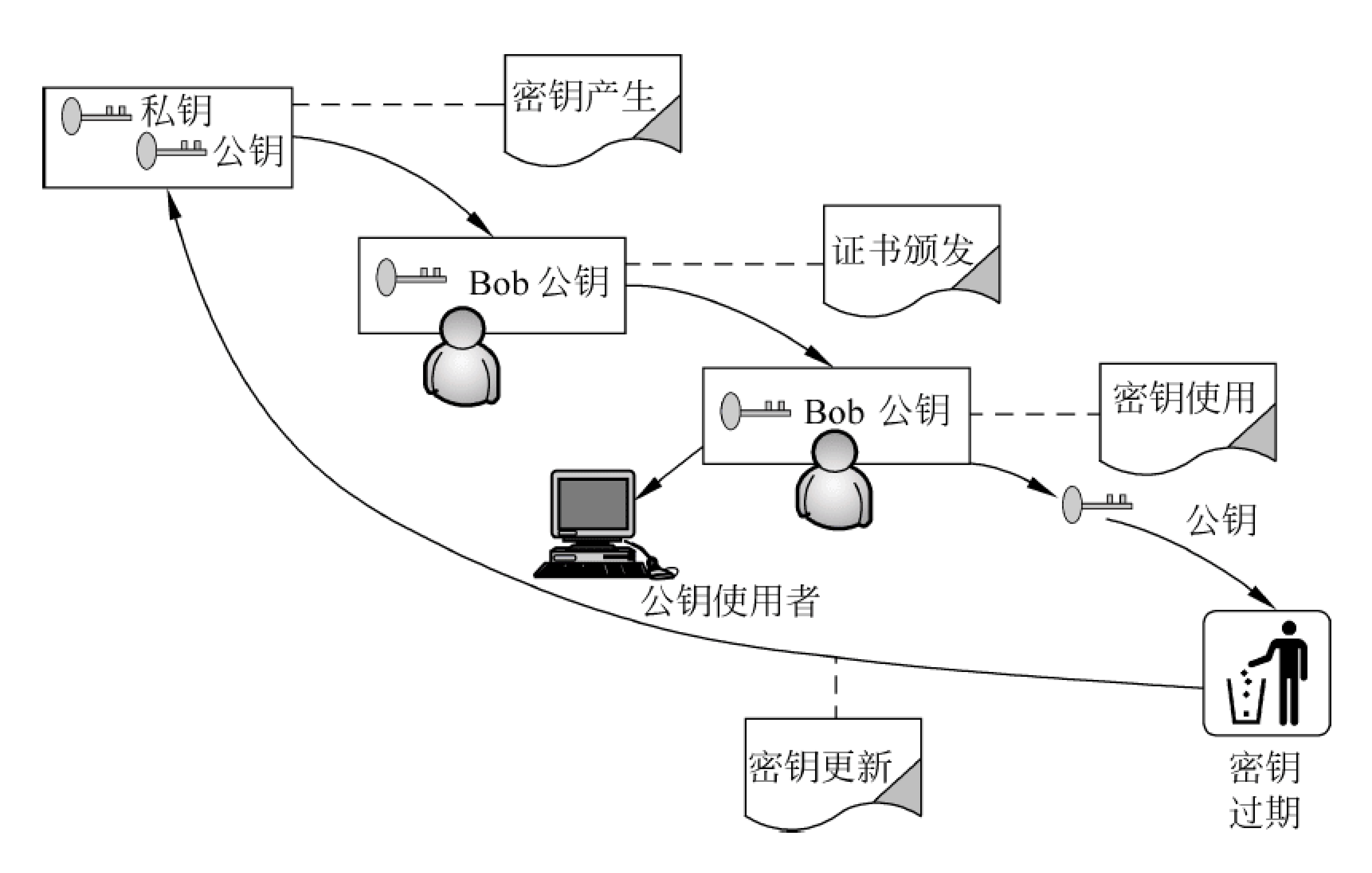


图 4.12　密钥生命周期

(1) 密钥产生

为了申请一个证书，需要为证书持有者产生一个公钥/私钥对。密钥对可以在终端实体注册之前或注册过程中产生。密钥对的生成可以在终端实体(证书的申请者)中执行，由端用户中支持 PKI 的应用程序完成；也可以由 RA 或 CA 产生(一般由一个 CA 附属但信息独立的密钥管理软件生成密钥对)；或者由一个可信的第三方产生。

如果是终端实体产生密钥对，则私钥使用用户密码进行保护，并存储在只有私钥拥有者才能够访问的安全存储区，公钥被加入证书申请中。

如果是密钥管理软件产生公私钥对，则终端实体或 RA 向 CA 提交证书签发请求时，不

提供公钥文件，而是指明由 CA 附属的密钥管理软件提供终端实体的公私钥对。由 CA 的密钥管理软件为终端实体产生公/私钥对，私钥暂时保存在密钥管理层，不进入 CA。CA 创建请求者的证书并对其进行签名后，以安全的方式将终端用户的私钥和对应的公钥证书一起分发给终端实体，然后销毁该私钥。这种情形下，终端用户的私钥、对应的公钥证书和解密该私钥的分割信息被保存在有效证书颁发列表中。

（2）密钥封装为证书

端用户的公钥连同其他信息被 CA 封装为数字证书并签名，然后将证书发送到指定的资源库中，例如其自身携带的数据库或系统指定的目录服务器中。

签名时，CA 从其密钥池中随机选取一对密钥，使用其中的公钥为终端实体签发证书（对证书进行数字签名）。证书签发成功后，CA 将证书和这对密钥的 ID 存入其有效证书颁发表中。最后，CA 根据证书的不同类型，把证书分别存放在其加密证书颁发表和签名证书颁发表中。

（3）密钥使用

用户从注册中心或其指定的服务器下载端用户的证书后，对该证书中的 CA 签名进行验证，同时查看证书有效期和证书持有者名称等信息后（即进行证书验证），即可以使用该公钥进行加密或签名验证等服务。

（4）密钥过期

密钥超过有效期后即过期。密钥过期将触发密钥更新过程。

（5）密钥更新

密钥过期后，需要从旧的密钥中产生新的密钥，这时进入密钥更新周期；同时，在证书未过期时用户提交证书更新请求也将触发密钥更新。密钥更新后将重新开始密钥生命周期。用户可以直接向 CA 提交证书更新请求，CA 也可以在一个合法的密钥对过期时（或接近过期时），自动进行新的公/私钥的自动产生和相应证书的颁发。密钥更新后，旧的证书进入“证书废止和撤销”阶段（见 4.6.3 节）。

4.6.3 证书生命周期

（1）证书申请

证书申请包括密钥对生成（见 4.6.2 节）和信息登记两部分。前者为端用户（证书申请者）生成并保存密钥对的过程，后者是为了申请证书，申请者向认证机构提交证书相关信息的过程，这些信息包括证书持有者的电子邮件、通用名、组织单位、组织和国家等。

端用户产生的私钥被保存在其本地安全存储空间，而公钥作为证书申请信息的一部分，连同其他信息被发送至 CA 进行证书申请。一般由用户通过支持 PKI 的应用程序，如 Web 浏览器向认证机构申请数字证书。

当用户密钥对由 CA 的密钥管理中心（或密钥管理软件）生成时，证书申请中可以不包

括用户的公钥,而只包括证书申请信息,公钥可从 CA 的密钥管理中心获得。

(2) 证书请求校验

端用户的证书请求发送到 CA 后,通常,CA 的注册中心 RA 验证用户的证书请求。校验的过程需要保证证书持有者的名称在整个 CA 中是唯一的,并且该名称是用户的真实名称。校验通过后,RA 将通过校验的用户证书请求提交给 CA。

(3) 证书生成

证书生成包括证书的创建和签名。用户的证书申请通过校验后,CA 处理该证书请求,利用证书申请中的部分登记信息,结合其他信息生成数字证书,并且使用自己的私钥对证书进行签名。

(4) 证书发布

认证中心在生成用户证书之后,会把用户的证书发送到指定的资源库中,如内部目录服务或公用服务器,以便证书的使用者获得该证书,并验证证书的合法性。

(5) 证书存储

端用户或证书的使用者从认证中心(或认证中心指定的公用服务器)下载证书后,应将证书保存在用户计算机的安全空间内。可以将证书保存在证书持有者计算机硬盘的秘密空间里,一般人无法存取,但可以通过应用程序提供的接口或浏览器导出,存储在证书使用者的计算机里以供使用和备份。

(6) 证书验证

不同于证书请求校验,证书验证是指证书的使用者(使用证书中的公钥进行加密或签名验证的用户)在给定的目标证书和一个可信密钥(信任锚)之间找到一个完整的路径,并且检查路径上每个证书的合法性,以确认该证书中的公钥是证书持有者的合法、有效的密钥。

以 CA 层次结构模型为例,假设域内所有证书均存放在同一个目录服务器中,则只要以目标证书的发布者(签发该证书的 CA)作为主题,在目录服务器中沿着证书信任链(CA 层次结构中的信任树)寻找颁发者的证书,直到找到发布者名字和证书颁发者名字相同的那张证书,即该域内的根证书。然后,系统验证证书路径上每个证书的合法性,包括证书是由可信 CA 签发的,进行证书的数据完整性验证,确认证书处在有效期内并且证书没有被撤销(不在证书废止列表 CRL 中)。此外,根据需要,还可以验证证书的使用方式和策略限制等。

(7) 证书废止和撤销

有多种原因可能导致证书被废止或撤销,如证书持有者的私钥损坏或丢失、密钥和证书过期、由于某种原因证书持有者向认证机构发出证书撤销请求等。

CA 撤销证书时,会把证书从其目录服务器和 CA 有效证书列表中删除,然后将其置于 CA 的归档证书列表中;如果该证书尚未过期(即证书持有者在证书未过期的情况下,向认证中心 CA 发出证书撤销请求),CA 还需要把该证书置于证书撤销列表 CRL 中。

4.7 PKI 相关标准

PKI 标准使得多个 PKI 系统之间可以交互，并且可以使不同应用程序面对单一的、固定的接口。如图 4.13 所示，PKI 体系的标准包括 ASN.1、X.509、LDAP、PKIX 和 PCKS 等。由这些标准定义和实现的 PKI 系统可以提供多种安全应用协议的支持，如 SSL/TLS、IPSEC/PPTP、S/MIME 和 SET 等，这些 PKI 应用协议具有其各自的协议标准，不属于 PKI 体系标准的规定范围。在这些应用协议之上，又可以提供多种安全服务或构建各种安全应用系统，如安全电子邮件、网上银行和虚拟专用网（VPN）等。本节给出 PKI 体系自身的相关标准，对于 PKI 应用协议及其相关标准将在 4.9 节简述，并且在本书后面的相应章节中详细描述其原理、应用和实施。

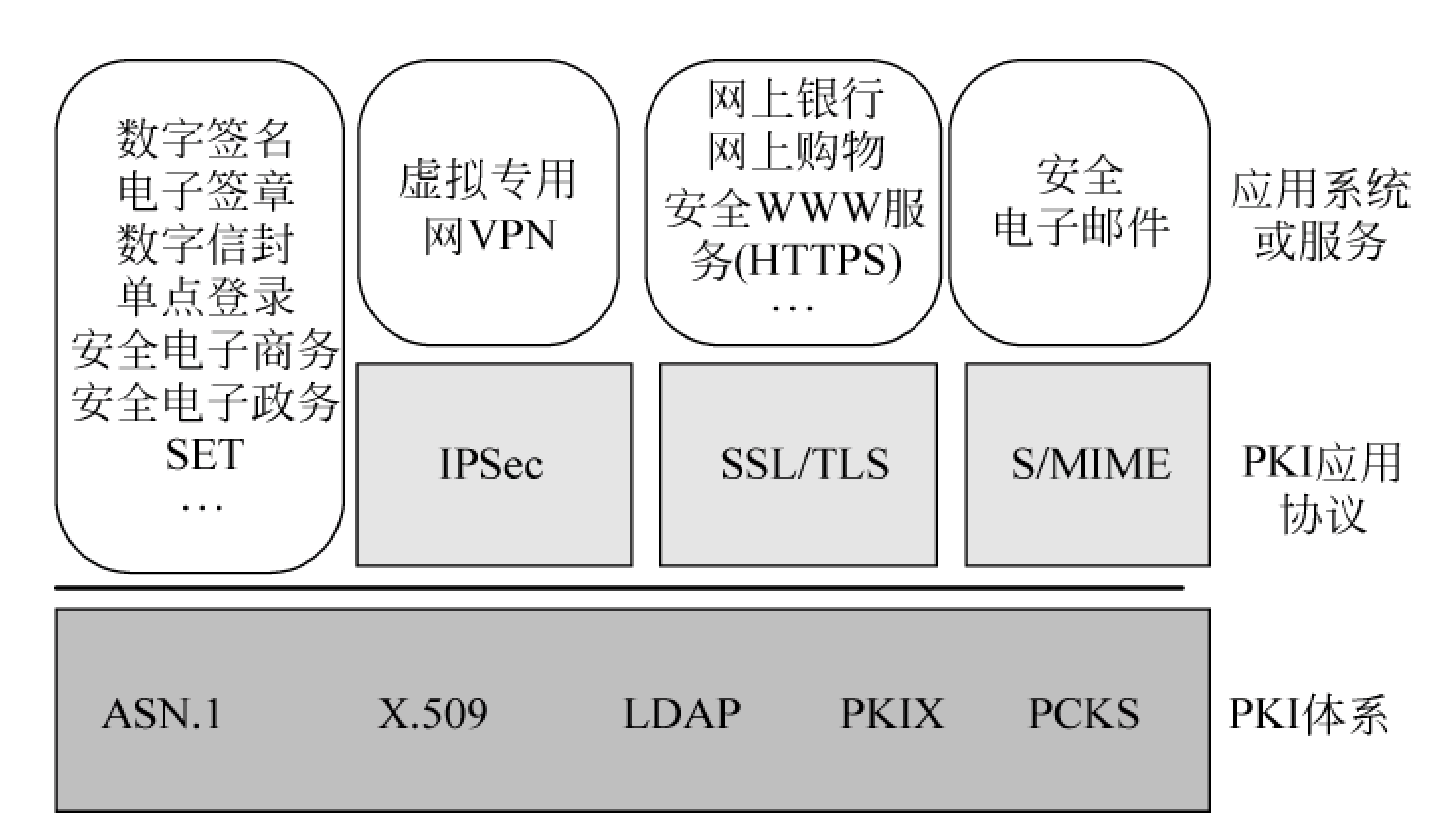


图 4.13 PKI 相关标准及应用体系

从整个 PKI 体系建立与发展的历程来看，PKI 体系自身主要包括以下一些标准。

1. ASN.1 基本编码规则

为了保证通过网络进行有效的数据传输，ISO 和 ITU-T 为 7 层网络协议中的应用层数据通信制定了一种标准的数据类型描述语言，称为抽象语法描述（abstract syntax notation.1，ASN.1）。

ASN.1 是 ISO 和 ITU-T 的联合标准，最初是 CCITT X.409：1984 的一部分。由于其广泛应用，1988 年 ASN.1 移到独立标准 X.208，1995 年进行全面修订后变成 X.680 系列标准。ASN.1 本身只定义了表示信息的抽象语法，但是没有限定其编码的方法。X.680 描述了它的基本语法规范，X.690 及其修正版对于按照 ASN.1 语法实现的几种编码方法和具体编码规则进行了规定。

各种 ASN.1 编码规则提供了由 ASN.1 描述其抽象语法的数据的值的传送语法（具体

表达)。标准的 ASN.1 编码规则有基本编码规则(basic encoding rules,BER)、规范编码规则(canonical encoding rules,CER)、唯一编码规则(distinguished encoding rules,DER)、压缩编码规则(packed encoding rules,PER)和 XML 编码规则(XML encoding rules,XER)。

目前在 PKI 的相关协议和应用中,所有数据标准、格式和结构定义均采用 ASN.1 语法进行描述,包括证书、黑名单、扩展项和管理协议等。几乎所有的 PKI 系统都是基于 ASN.1 和 DER(distinguised encoding rules for ASN.1)实现的,它已成为支持 PKI 和公钥密码体制的应用系统之间互通的桥梁。

ASN.1 主要用于证书的语法及编码规则描述,它并不规定证书的字段内容,证书的字段由 X.509 定义。

例如,X.509 公钥证书格式的 ASN.1 的部分语法描述如下:

```
Certificate::= SIGNED{SEQUENCE{
  version[0]
  serialNumber
  signature
  issuer
  validity
  subject
  subjectPublicKeyInfo
  issuerUniqueIdentifier
  …
  extensions}}
```

2. X.500

X.500 是一套已经被国际标准化组织(international organization for standards,ISO)接受的目录服务系统标准,它定义了一个机构如何在全局范围内共享其名字和与之相关的对象。X.500 是层次性的,其中的管理域可以提供这些域内的用户和资源信息。

X.500 实际上不是一个协议,它由一个协议族组成:X.501 模型强调目录服务基本模型和概念;X.509 认证框架规定如何在 X.500 中处理目录客户和服务器的认证;X.511 抽象服务定义 X.500 提供的功能性服务;X.518 分布式操作过程表明如何跨越多台服务器处理目录服务;X.519 包括目录访问协议(directory access protocol,DAP)、目录系统协议(directory system protocol,DSP)、目录操作绑定协议(directory operator protocal,DOP)和目录信息映像协议(directory information shadowing protocol,DISP)。

在这些 X.500 标准中主要定义有多种内容。一个信息模型:确定目录中信息的格式和字符集,如何在项中表示目录信息(定义对象类、属性等模式);一个命名空间:确定对信息进行的组织和引用,如何组织和命名项——目录信息树(directory information tree,DIT)和层次命名模型;一个功能模型:确定可以在信息上执行的操作;一个认证框架:保证目录中信息的安全,如何实现目录中信息的授权保护——访问控制模型;一个分布操作模型:

确定数据如何进行分布和如何对分布数据执行操作，如何将全局目录树划分为管理域进行管理——目录管理模型，客户端与服务器通信的协议——目录访问协议，将用户请求在服务器之间进行链接所需的目录系统协议，将选定的信息在服务器之间进行复制所需的目录信息映像协议（directory information shadowing protocol，DISP），用于自动在服务器之间协商连接配置的目录操作绑定协议 DOP。

X. 500 虽然是一个完整的目录服务协议，但在实际应用的过程中，却存在着不少障碍。由于目录访问协议是严格遵照复杂的 ISO7 层协议模型制定的，对相关层协议环境要求过多，主要运行在 Unix 机器上，在许多小系统上，如 PC 和 Macintosh 上无法使用，因此没有多少人按照 DAP 开发应用程序，TCP/IP 协议体系的普及，更使得这种协议越来越不适应需要。

在 PKI 体系中，一般采用 X. 500 建议的目录服务器作为数字证书、CRL 和黑名单等的发布。但由于 X. 500 实施复杂，在 PKI 的实际应用中，一般采用 X. 500 的简化版本——LDAP 来实现证书的发布和查询服务。

3. X. 509

X. 509 是由国际电信联盟（ITU-T）制定的数字证书标准。在 X. 500 确保用户名称唯一性的基础上，X. 509 为 X. 500 用户名称提供了通信实体的鉴别机制，并规定了实体鉴别过程中广泛适用的证书语法和数据接口。这一标准的最新版本是 X. 509 v3，它定义了包含扩展信息的数字证书。该版数字证书提供了一个扩展信息字段，用来提供更多的灵活性及特殊应用环境下所需的信息传送。X. 509 标准定义的证书是目前普遍使用的证书格式。

4. PKIX

如前所述，在 X. 509 标准的基础上，IETF PKIX 工作组制定了支持 X. 509 PKI 的系列协议标准，即 PKIX，它定义了在 Internet 上利用 X. 509 证书实现 PKI 体系结构的具体操作细节和规范。典型的协议及其标准如下。

（1）证书和 CRL 标准

为了在全球范围内通过 Internet 建立基于 X. 509 v3 的公钥证书体系，IETF 的 PKIX 工作组开发了一系列证书和 CRL 相关标准，其中最重要的是 PKIX 证书和 CRL 大纲（RFC 2459 和 RFC 3280：PKIX certificate and CRL profile，后者是对前者的更新）。这些文档描述了 X. 509 v3 公钥证书的内容及其扩展项的取值范围、PKI 证书和 CRLs 的格式和语法，以及在 Internet 环境中证书路径处理的步骤，还给出了可用于所有数据结构的 ASN. 1 模块等。

其他主要相关协议标准包括如下内容。

• RFC 4630[Update to DirectoryString Processing in the Internet X. 509 Public Key

Infrastructure Certificate and Certificate Revocation List(CRL)Profile]：PKIX 证书和 CRL 中的目录字符串处理更新。此文档描述了对 PKIX 证书和 CRL 中目录字符串操作的更新，其中，UTF8 字符串和可打印的字符串是首选的编码方式。

- RFC 3039[Internet X.509 Public Key Infrastructure Qualified certificates]：PKIX 资格证书。为了从法律上支持数字签名，必须引入一些基本要求以限制公钥证书，这种证书称为资格证书。1998 年 12 月，IETF PKIX 工作组将 RFC 2459 进一步改进使 PKIX 公钥证书能够支持资格证书，这些改进在 RFC 3039 中进行了描述。
- RFC 3281[An Internet Attribute certificate profile for authorizations]：支持授权的属性证书大纲。按照 ITU 的定义，属性证书不同于公钥证书，它是指对用户属性和其他信息使用 CA 私钥进行签名而形成的证书。和 X.509 公钥证书类似，属性证书也是一个复杂的结构，包括基本字段和一系列可扩展项。RFC 3281(An Internet Attribute certificate profile for authorizations)中的属性证书大纲描述了属性证书的内容及其扩展项，同时定义了 X.509 属性证书需要支持授权服务时所采用的数据格式。

(2) 在线证书状态协议(OCSP)

OCSP(online certificate status protocol)是 IETF 颁布的用于检查数字证书在某一交易时间是否有效的标准。在 OCSP 之前，用户只能使用下载和处理证书撤销清单 CRL 的方式检查证书的有效性，而 CRL 方式存在效率低、网络资源消耗大等缺点。OCSP 为 PKI 提供了一种比下载和处理 CRL 的传统方式更快、更方便和更具独立性的检验数字证书有效性的途径。OCSP 请求是独立于协议的(HTTP 是通用的方式)。

与 OCSP 相关的 PKIX RFC 文档包括 RFC 2560(X.509 Internet public key infrastructure online certificate status protocol)以及 RFC 5019(The Lightweight Online certificate status protocol profile for high-volume environments)。前者定义了使用 OCSP 进行证书状态查询的应用程序(OCSP 客户端)和 OCSP 服务器之间进行消息交换的数据格式，包括 OCSP 请求和应答消息以及各种消息的语法。后者给出了大型 PKI 环境下的轻量级在线证书状态协议，此协议用于大型 PKI 环境下的在线证书状态的确认，也适用于需要轻量级解决方案来减小通信带宽及客户端进程的 PKI 环境。

(3) PKIX 操作协议

X. 509 PKI 体系中的相关实体通过 PKIX 操作协议获得证书、CRL 或证书状态等信息，PKIX 操作协议规定了通过 DNS、LDAP、FTP、TCP/IP、HTTP 和 MIME 等协议来发布和查询证书或 CRL 等的具体过程。OCSP 也属于 PKIX 操作协议的一部分，其他主要 PKIX 操作协议包括：

- RFC 2559 [PKIX operational protocols-LDAPv2]：描述使用 LDAPv2 作为 PKI 实体发布和获取证书及 CRL 的协议。
- RFC 2587[PKIX LDAPv2 Schema]：定义了一个最小模型支持使用 LDAPv2 获取

公钥证书和CRL。

- PKIX Operational Protocol - LDAPv3<draft-ietf-pkix-ldap-v3-05.txt>：是一个IETF草案，描述了LDAPv3版本中，支持基于X.509证书和CRL所需要的一些新特征。
- RFC 2585[Internet X.509 Public Key Infrastructure Operational Protocols：FTP and HTTP]：FTP和HTTP的PKIX操作协议。这个协议是一组采用X.509和CRLs的标准Internet公钥基础设施的一部分，描述了使用FTP和HTTP从PKI证书库中获得证书和CRLs的标准。
- RFC 2875[Diffie-Hellman Proof-of-Possession Algorithms]：描述了使用D-H密钥交换算法在类似PKCS#10的证书请求中用于数据完整性检查和数据拥有者证明的方法。

(4) PKIX管理协议

PKIX管理协议定义了X.509 PKI用户和其管理实体之间在线交互时使用的各种管理协议，管理协议分为两部分，一部分定义了管理协议传输的信息格式，另一部分定义了控制这些信息传递的传输协议。主要包括如下相关协议。

- RFC 2797[Certificate Management Messages over CMS]：CMS证书管理协议。利用CMS来进行证书管理的协议，它列出了两项在Internet PKI中的即时需求：对公钥证书产品和基于CMS的服务的接口的需求，对采用D—H公钥算法进行DSA证书签名的证书登记协议的需求。
- RFC 4210[Internet X.509 Public Key Infrastructure Certificate Management Protocols]：PKIX证书管理协议。此协议描述了PKIX的证书管理(CMP)。证书管理为PKI组件之间提供了在线交互功能，包括CA和客户端系统的交互。
- RFC 4211[Internet X.509 Public Key Infrastructure Certificate Request Message Format(CRMF)]：PKIX证书请求消息格式。此文档描述了证书请求消息的格式和语法。此语法用于发送一个证书请求给CA(可能通过RA)以获得X.509的证书产品。

其他协议和标准包括证书管理政策和证书操作规范、提供抗抵赖的时间戳和数据认证服务等。

5. 轻量级目录访问协议

轻量级目录访问协议(lightweight directory access protocol，LDAP)是一个Internet协议，用来存取基于X.500的目录服务，LDAP规范简化了X.500目录访问协议，并且在功能性、数据表示、编码和传输方面都进行了相应的修改。1997年，LDAPv3成为互联网标准，由IETF LDAPext来定义和进行标准化。

LDAP协议从1993年批准，产生了LDAPv1版本，随后于1997年发布了第三个版本

LDAPv3，它的出现是 LDAP 协议发展的一个里程碑性标志，它使 LDAP 协议不仅仅作为 X. 500 的简化版，同时提供了 LDAP 协议许多自有的特性，使 LDAP 协议功能更为完备，具有了更大的生命力。

LDAPv3 是一个协议族。

- RFC 2251：LDAPv3 核心协议，定义了 LDAPv3 协议的基本模型和基本操作。
- RFC 2252：定义了 LDAPv3 中的基本数据模式(Schema)(包括语法、匹配规则、属性类型和对象类)以及标准的系统数据模式。
- RFC 2253：定义了 LDAPv3 中的分辨名(DN)表达方式。
- RFC 2254：定义了 LDAPv3 中的过滤器的表达方式。
- RFC 2255：LDAP 统一资源地址的格式。
- RFC 2256：在 LDAPv3 中使用 X. 500 的 Schema 列表。
- RFC 2829：定义了 LDAPv3 中的认证方式。
- RFC 2830：定义了如何通过扩展使用 TLS 服务。
- RFC 1823：定义了 LDAP 客户端 API 开发接口。
- RFC 2847：定义了 LDAP 数据导入、导出文件接口。

在这些协议中，主要定义了 LDAP 的内容，同时主要定义了一个信息模型：确定 LDAP 目录中信息的格式和字符集，如何表示目录信息(定义对象类、属性、匹配规则和语法等模式)；一个命名空间：确定对信息进行的组织方式——目录信息树，以 DN(distinguished name)和 RDN(relative distingushed name)为基础的命名方式，以及 LDAP 信息的 Internet 表示方式；一个功能模型：确定可以在信息上执行的通信协议以及在客户端进行这些操作的 API(application programming interface)接口；一个安全框架：保证目录中信息的安全，匿名、用户名/密码、SASL(simple authentication and security layer)等多种认证方式，以及与 TLS 结合的通信保护框架；一个分布式操作模型：基于 Referral 方式的分布式操作框架；一个 LDAP 扩展框架：基于控制和扩展操作的 LDAP 扩展框架。

LDAP 目录存储的数据不是关系型数据库，LDAP 条目组织一般按照地理位置和组织关系进行组织，非常直观。LDAP 可提供快速响应和大容量查询服务，并且提供多目录服务器的信息复制功能，并且允许根据需要使用访问控制列表 ACL(access control list)控制对数据读和写的权限。

大多数的 LDAP 服务器都为读密集型的操作进行专门的优化。因此当从 LDAP 服务器中读取数据的时候会比从关系型数据库中读取数据快一个数量级。同时因为没有对写操作的性能进行任何优化，所以大多数 LDAP 目录服务器并不适合存储需要经常改变的数据。LDAP 的这些特性使得其非常适合用来存储数字证书。目前，LDAPv3 已经在 PKI 体系中被广泛应用于证书信息发布、CRL 信息发布、CA 政策以及与信息发布相关服务，提供数字证书的存储、发布和查询服务。

6. 公钥密码学标准 PKCS

PKCS(public-key cryptography standard)是由美国 RSA 数据安全公司及其合作伙伴制定的一组公钥密码学标准，定义了公开密钥的加密、交换、数据加密、签名、证书请求、密钥存放和多种新兴加密算法标准，其中大多数被广为应用，成为事实上的工业标准，其中包括证书申请、证书更新、证书作废表发布、扩展证书内容以及数字签名、数字信封的格式等方面的一系列相关协议。到 1999 年年底，PKCS 已经公布了以下标准。

- PKCS#1：定义 RSA 公开密钥算法加密和签名机制，主要用于组织 PKCS#7 中所描述的数字签名和数字信封。
- PKCS#3：定义 Diffie-Hellman 密钥交换协议。
- PKCS#5：描述一种利用从密码派生出来的安全密钥加密字符串的方法。使用 MD2 或 MD5 从密码中派生密钥，并采用 DES—CBC 模式加密。主要用于加密从一个计算机传送到另一个计算机的私人密钥，不能用于加密消息。
- PKCS#6：描述了公钥证书的标准语法，主要描述 X.509 证书的扩展格式。
- PKCS#7：定义一种通用的消息语法，包括数字签名和加密等用于增强的加密机制，PKCS#7 与 PEM 兼容，所以不需其他密码操作，就可以将加密的消息转换成 PEM 消息。
- PKCS#8：描述私有密钥信息格式，该信息包括公开密钥算法的私有密钥以及可选的属性集等。
- PKCS#9：定义一些用于 PKCS#6 证书扩展、PKCS#7 数字签名和 PKCS#8 私钥加密信息的属性类型。
- PKCS#10：描述证书请求语法。
- PKCS#11：定义了一套独立于技术的程序设计接口，用于智能卡和 PCMCIA 卡之类的加密设备。
- PKCS#12：描述个人信息交换语法标准。描述了将用户公钥、私钥、证书和其他相关信息打包的语法。
- PKCS#13：椭圆曲线密码体制标准。
- PKCS#14：伪随机数生成标准。
- PKCS#15：密码令牌信息格式标准。

在 PKI 系统中，PKCS 标准被广泛运用于证书申请、数据加密和签名、密钥存储、密钥传输等。例如，在 PKI 系统中，端用户的证书申请应符合 PKCS#10 标准；使用密码保护用户的 RSA 私钥采用 PKCS#8 定义的方式实现；公/私钥和相应的证书通常保存在一个应用指定的密钥存储中，这些密钥存储通常提供一个标准 API 接口，以便使用 API 的其他应用可以共享对相同密钥资料的访问，在 Netscape 和许多 UNIX 平台的应用中，这个接口符合 PKCS#11 标准。

和 X.509、LDAP 一样，PCKS 被 PKIX 体系结构和系列标准所支持。

7. WPKI

从原理上说，无线 PKI 和 Internet PKI 没有根本区别，同样可以建立基于 X.509 证书的 PKI 体系来保证无线安全。但在目前的无线应用中，必须考虑到无线终端（手机、手持终端等）的容量小、处理能力低以及无线网络的带宽比 Internet 窄等多种因素，因此要求尽量减小证书数据长度，以节省终端存储资源和无线带宽资源；尽量减少证书处理难度，减轻终端对处理能力的要求，提高效率。

WPKI 目前有比较多的体系，但多数是企业标准，真正开放的标准目前只有 WAP Forum 制定的 WAP（wireless application protocold）协议。WAP 的目标就是通过这种技术，使 Internet 的内容和各种增值服务适用于手机用户和各种无线设备用户，并促使业界采用这一标准。

WTIS 类似于 TCP/IP 中的 SSL，但由于手机及手持设备的处理和存储能力有限，WAP 论坛在 TLS 的基础上做了简化，提出了 WTLS 协议（wireless transport layer security），以适应无线的特殊环境。

WPKI 既考虑到了对已有标准的利用，同时，也针对无线应用特点，增加了新的证书类型：WTLS 压缩证书和证书 URL，同时对 X.509 也进行了一些限制，以便应用于无线环境。

与 X.509 证书相比，为了适应无线应用环境，WAP 协议中对 X.509 证书进行了一些限制，如证书序列号建议不要超过 8B(63 位，最高位不能置 1)；缺少了 subjectUniqueIdentifier 和 issuerUniqueIdentifier 两个可选项；签名算法指定了两种：sha1WithRSAEncryption 和 ecdsawith-SHA1；公钥也只定义了两种：rsaEncryption 和 id-ecPublicKey；RSA 算法的密钥强度不能小于 1024 位，椭圆曲线算法的密钥强度不能小于 160 位。同时，协议中规定，系统必须支持的标准扩展项包括：keyUsage，extKeyUsage，certificatePolicies，subjectAlt_Name 和 basicConstraints。

WTSL 协议中，定义了不同于 Internet X.509 的证书格式，它的证书大小约是 X.509 证书的 40%，而且在 WAP1.3，即 WPKI 的许可下，将会减少客户/服务器身份鉴别的负担，但 WTLS 建立的安全连接是在 WAP 网关和手持设备之间，WAP 网关和 Web Server 之间如果也要保密，则需要考虑另外的方式，即在这种模型中无法实现端到端的加密。

目前 PKI 体系中已经包含了众多的标准和相关协议，由于 PKI 技术的不断进步和完善，及其应用的不断普及，将来还会有更多的标准和协议加入。

4.8 成熟 PKI 系统简介

PKI 的应用在国外已经有很多年，在商业和政府中有着成熟稳定的应用。本节介绍商业应用中比较成熟的 VeriSign、Entrust 等系统，同时也介绍政府应用中比较成熟的美国联

邦 PKI(FPKI)和加拿大政府 PKI(GOC PKI)。

4.8.1 商业应用

1. VeriSign

VeriSign 是目前全球最大的数字信任服务提供商，总部位于美国加利福尼亚的山景(Mountain View)。VeriSign 公司源于 RSA 公司的认证服务中心 CSC(certification service center)。该中心最初是为各企业构建 PKI/CA。1995 年，CSC 从 RSA 公司分离出来成立了 VeriSign。VeriSign 的 CA 系统在近 50 个国家和地区运行，其产品 Onsite 应用广泛，具有巨大的用户群体，包括波音公司、通用电气和菲利浦等。

信任服务和支付服务是 VeriSign 数字认证方面的两个主要服务项目。VeriSign 通过其网站以及众多的 ISP，向全球范围内的网站、软件开发商和个人提供信任服务，这其中包括签发专门应对网站鉴别和加密的 SSL 服务器证书。目前，包括亚马逊、雅虎购物、美国在线在内的全球 26 万余家网站安装了 VeriSign 的 SSL 服务器证书，VeriSign SSL 服务器证书在全球的市场占有率高达 90%以上。VeriSign 的支付服务可以使客户安全地处理和管理在线支付活动，全球 6 万余家商户接受了该项服务。另外，VeriSign 还针对无线电子商务提供了一整套的 PKI 解决方案。

为了扩展自己的服务领域和范围，VeriSign 与 American Express、Checkpoint、Microsoft、RSA 建立了战略合作关系，包括英国、法国、德国、意大利、澳大利亚、巴西、南非、中国、日本和韩国等几十个国家和地区在内的 48 家数字信任服务提供商加入了 VeriSign 信任网络。作为 VeriSign 在中国大陆的首要合作伙伴，国内的天威诚信公司 2001 年加入了 VeriSign 信任网络。

VeriSign 通过其基本产品 Onsite 来实现信任服务。VeriSign 的证书产品有安全电子邮件证书、SSL 服务器证书、设备证书(如 VPN、Cable Modem)、WAP 证书和代码签名证书等。

OnSite 是一个完整的 PKI 产品，被用来对企业内部互联网、外部网、VPN 以及电子商务应用软件提供具有最大限度的灵活性、有效性和可扩展性的安全服务，它可以帮助您高效地建立一个健壮的、满足需求的 PKI 系统。与纯软件的解决方案和完全由自己建立 PKI 系统不同，OnSite 可以使用户根据自己的实际情况灵活地配置 PKI 的规模，充分发挥自己的特长。Onsite 提供了安全 Web 访问、本地主机、密钥管理和恢复、证书确认、应用程序工具包、双钥支持和自动证书恢复等功能。

2. Entrust

Entrust 起源于加拿大北方电信(Nortel)的安全网络部门，总部设在美国得克萨斯州。该部门最初也是为企业提供 PKI 解决方案，他们的产品称为 Entrust。1997 年 2 月，安全网络部门从北方电信分离出来，成立了 Entrust Technologies。

Entrust 主要的业务是为企业提供 PKI 产品。1999 年 5 月，Entrust 通过 entrust. net 开始提供面向机构的公共 Web Server 证书服务。

Entrust 公司的 PKI 产品是 Entrust/PKI 5.0。Entrust 的 CA 可以向各种设备或应用程序颁发数字证书，包括终端 PC 用户、Web 服务器、Web 浏览器、VPN 设备和 SET 用户等。凡是支持 X.509 证书格式的设备或应用程序都可以获得数字证书。此外，Entrust 的 CA 还可以针对个别特殊用户定制相应的特殊证书，并在这个特别证书里赋予该用户一些特殊权力。

PKI 5.0 CA 有较完善的 CA 数据库功能，包括数据库加密、完整性检验、CA 专有硬件、对敏感操作的分级权限设定等。PKI 5.0 也提供了较为完善的密钥备份和恢复系统，支持所有的证书撤销格式和标准，包括证书撤销列表(CRL)、CRL 分布点以及在线证书状态协议 OCSP(online certificate status protocol)。Entrust PKI 5.0 支持国际上的各项标准，如 X.509 格式证书、CRL、OCSP、LDAP、PKIX、PKCS、IPSec 和 SSL 等。

4.8.2 政府应用

1. 美国联邦 PKI(FPKI)

美国联邦 PKI(FPKI)是由 1996 年成立的美国联邦 PKI 筹委会与联邦首席信息官委员会共同研究、建立的，用于解决当时美国国内不同信任域之间的互操作性问题，使信任域不仅局限在本信任域环境中，而且可以使信任域扩展到整个联邦政府甚至是全球。联邦 PKI 支持在开放的网络如 Internet 上进行安全的交易，用于保障电子政务、电子采购的信息安全和实现对关键网络设备的保护，使得美国联邦机构可以与其他联邦机构、各级政府、私有性质的贸易伙伴、公众机构之间进行安全的电子交易。

美国联邦 PKI 体系主要由联邦桥 CA(federal bridge CA，FBCA)、首级 CA(principal CA，PCA)、次级 CA(subordinate CA，SCA)等组成，其体系结构如图 4.14 所示。

从图 4.14 中可以看出，联邦 PKI 体系中没有用根 CA 的概念，取而代之的是首级 CA，这是因为在美国，信任域的结构是多种多样的，联邦 PKI 体系可以支持层级(树状)结构、网状结构和信任列表等，而只有层级(树状)结构中的首级 CA 才称作根 CA。因此它允许加入联邦 PKI 体系中的机构可以使用任何结构的 PKI 信任域。联邦的桥 CA 是联邦 PKI 体系中的核心组织，是不同信任域之间的桥梁，它将不同类型的 PKI 结构连接在一起，实现彼此之间的信任，并将每一个单独的信任

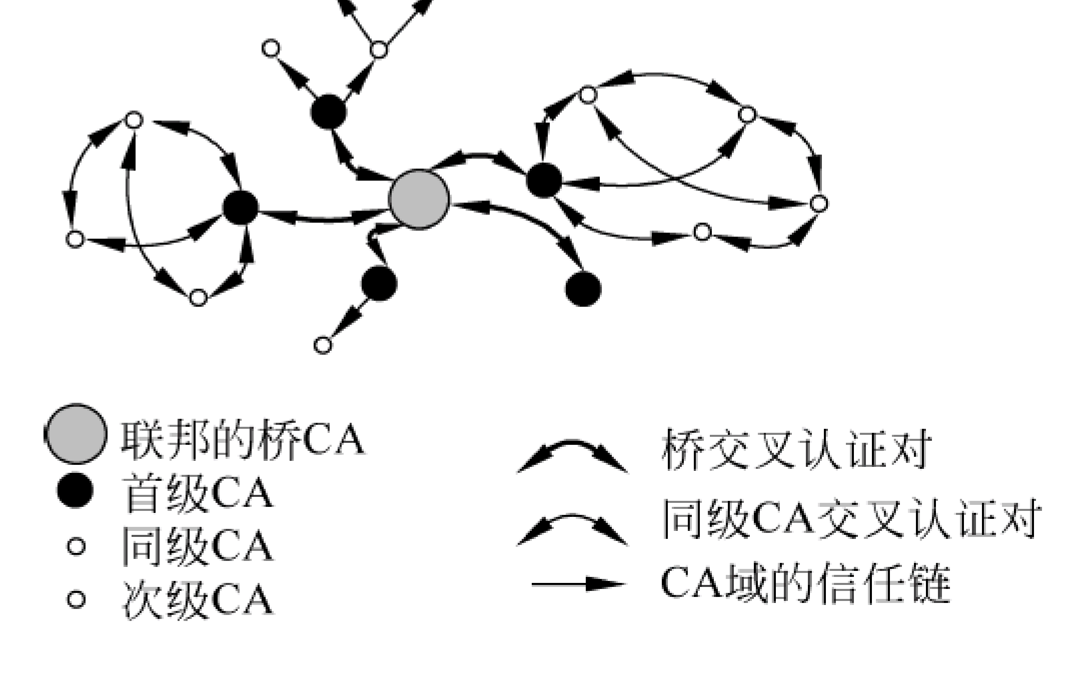


图 4.14 美国联邦 PKI 的体系结构

域通过联邦的桥 PKI 扩展到整个联邦 PKI 体系中。联邦的桥 CA 主要负责为不同信任域的首级 CA 颁发交叉认证的证书，建立各个信任域的担保等级与联邦的桥 CA 的担保等级之间的一一映射关系，更新交叉认证证书，发布交叉认证证书注销黑名单。但是它不要求一个机构在与另一个机构发生信任关系时必须遵循联邦 PKI 所确定的这种映射关系，而是可以采用它认为合适的映射关系确定彼此之间的信任。

目前，美国政府联邦 PKI 体系还处于研究和测试阶段，已经完成了对联邦 PKI 体系中安全电子邮件的数字签名的测试。

2. 加拿大政府 PKI(GOC PKI)

加拿大对于政府 PKI 体系的研究早于美国，在 1993 年加拿大通信安全部(communications security establishrnent，CSE)就已经开始了政府 PKI 体系的研究工作，当时主要是开发一种满足政府需求的 PKI 产品，实现无货架商业贸易。随后，陆续有部分联邦政府机构参入了 GOC PKI 体系的开发工作。经过若干年的研究，在 2000 年，加拿大政府在建立一个开放的 PKI 体系方面获得重要的进展，实现了联邦政府与公众机构、商业机构等进行电子数据交换时的信息安全保障，从而推动了政府内部管理的信息化进程。加拿大政府 PKI 体系结构如图 4.15 所示。

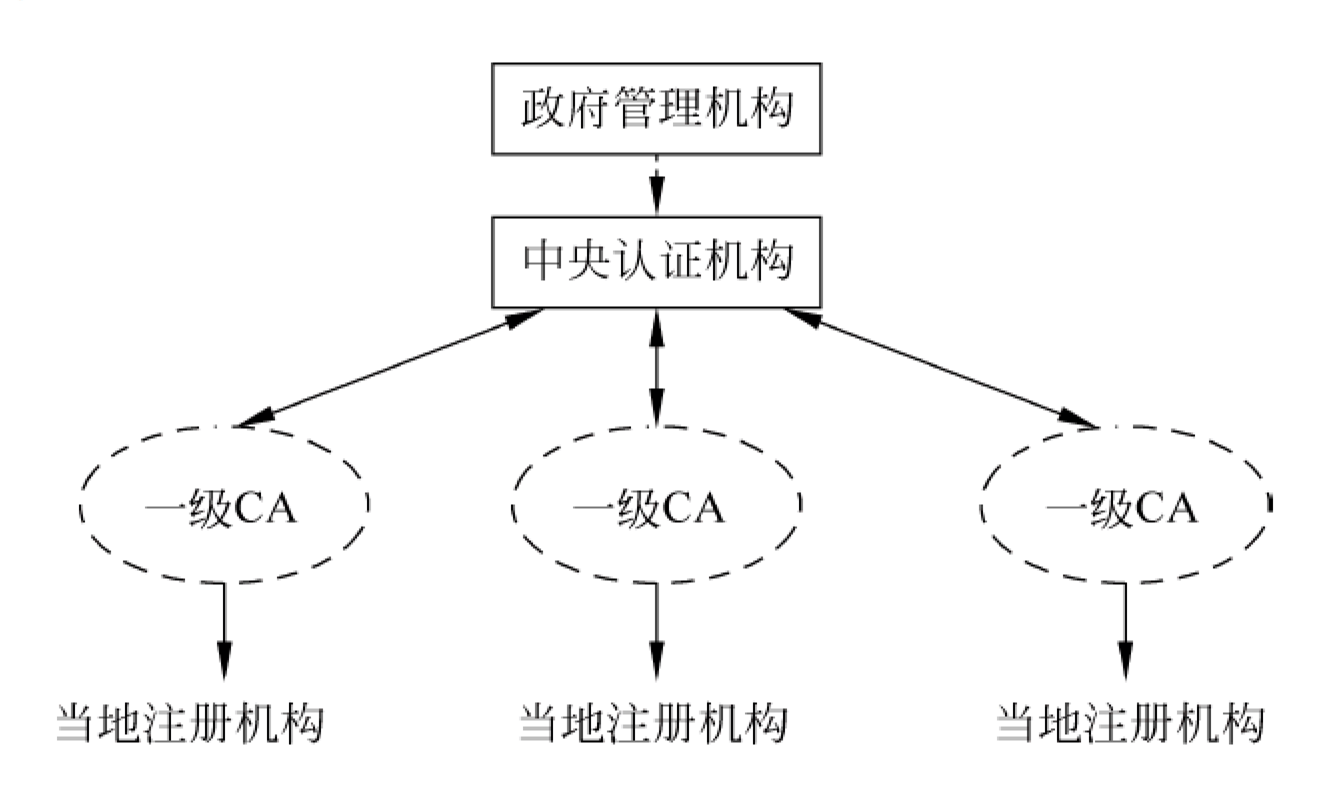


图 4.15　加拿大政府 PKI 体系结构

如图 4.15 所示，加拿大政府 PKI 体系是由政策管理机构(policy maragement authority，PMA)、中央认证机构(the caradian central facility，CCF)、一级 CA 和当地注册机构(local registration authority，LRA)组成。

其中 PMA 是一个若干部门共同组建的机构，由加拿大政府财政部秘书处领导，为政府 PKI 体系提供全面的政策指导，负责监督和管理加拿大政府 PKI 体系的政策实施情况。

CCF 是中央认证机构，它实施政府 PKI 体系中的所有策略，签署和管理与一级 CA 交叉认证的证书。

一级 CA 是由政府运营，制定一个和多个证书担保的等级，分发和管理数字证书，定期颁布证书注销黑名单。

当地注册机构是一级 CA 设置的登记机构或个人，其职责是认证和鉴别申请者的身份；为密钥恢复或证书恢复请求进行审批；接受并审批证书的注销请求。

加拿大政府 PKI 体系是一个完全政府行为的公开密钥体系结构，它充分考虑了交易的私有性和安全性，并把保护交易私有性和安全性列为信息高速公路的首要问题。它提供了完全一致的密钥管理办法，并为加密和数字签名提供了完全相同的验证过程，是数字认证、鉴别和智能卡等多项技术的集成。

加拿大政府 PKI 体系的信任域全部采用树状结构，可以快速地实现信任关系的查找，建立起信任关系。另外，树状结构的信任域间建立信任关系必须通过中央认证机构，不允许一个树状结构与另一个树状结构直接发生信任关系，中央认证机构是与外界建立信任关系的唯一接口，因此结构简单，易于操作，是一个值得借鉴的 PKI 体系。

4.9 PKI 实施与应用案例

4.9.1 小型 PKI 和 CA 设计案例

这种小型 PKI 系统不提供公共服务，一般在局域网内部使用，为内部用户或应用提供身份认证或加密所需的数字证书服务。

1. 系统需求

企业内部网络和外部网络相对独立，外部用户无法访问内部机密信息。内部网络运行电子邮件服务和 WWW 服务，需要提供签名电子邮件和 HTTPS 等安全应用，需要为内部网络的这些安全应用设计独立的 PKI 系统，要求 PKI 系统实现如下功能。

- 证书申请。
- 申请人身份验证。
- 证书发放和查询。
- 证书撤销。
- 证书更新。
- 密钥恢复。

2. 系统总体设计

如图 4.16 所示，PKI 系统由终端实体、CA、RA 和目录服务构成。为 CA 和 RA 设计管理客户端，实现证书和系统相关的管理功能。终端实体和管理客户端采用 HTTPS 与 CA 和 RA 进行交互。

系统采用单 CA 信任模型，为所有内部用户设立唯一的 CA 作为信任锚。

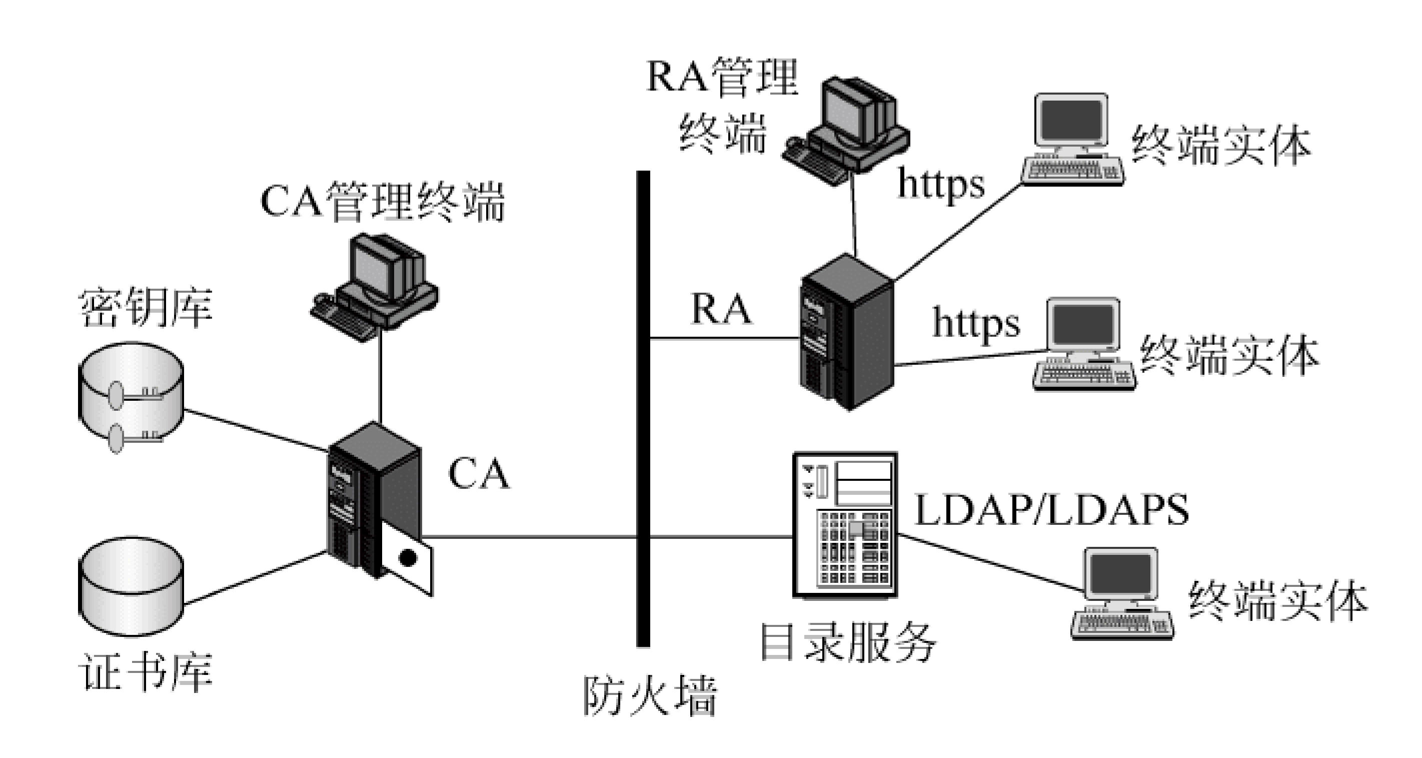


图 4.16　小型 PKI 体系结构示意图

由于内部网络相对可信，使用集中式生成密钥和证书的方式，使用密钥管理软件在 CA 服务器端为用户生成密钥对，然后采用网络或手工复制的方式将密钥及证书分发给使用者。

系统利用 OpenSSL 实现 CA 的证书功能，利用 OpenLDAP 存储用户证书和 CRL，并提供证书和 CRL 查询服务。

PKI 系统中的主要实体按图 4.16 所示方式设计。

- 终端实体：由 Web 客户端程序实现，通过浏览器和 PKI 系统交互。用户填写各种电子表单，包括证书申请、证书撤销申请、密钥恢复申请等。这些交互信息通过 HTTPS 保证其安全性。
- CA：通过一个(plypertext preprocessor，PHP)服务器端程序实现。利用 OpenSSL 的 PKCS 和 X.509 实现，通过其提供的 API 实现密钥对、证书和 CRL 的生成。
- RA：由一个 PHP 服务器端程序和 RA 管理员构成。其中 RA 服务器端程序接受终端实体的证书申请请求，并将其保存在数据库中。RA 管理员通过人工审核的方式验证证书申请人身份的有效性。
- 证书和密钥存储：证书、CRL 以及用户密钥对保存在数据库中，同时 LDAP 服务器保存证书和 CRL 副本。
- LDAP 服务：利用 OpenLDAP 实现 LDAP 服务器功能，通过其提供的 API 实现证书和 CRL 的存储、发布和查询等。

3. 系统运行和开发环境

(1) 运行环境

- 操作系统：Linux。
- Web 服务器：Apache。

(2) 开发工具和软件包

- PHP：利用其服务器端程序实现 CA 和 RA 服务器。
- OpenSSL：实现密钥对生成、证书生成和签发、CRL 生成等证书管理功能。

- OpenLDAP：使用 Linux 平台下的 OpenLDAP 服务器存储证书和 CRL。

(3) 数据库

采用 SQLServer 数据库存储证书申请及其他相关信息。

4. 主要功能的设计和实现

(1) 证书申请

终端实体通过 Web 浏览器向 RA 服务器发送证书申请请求，RA 服务器端程序通过 HTTPS 方式接受用户的证书申请请求，并将证书申请信息保存在数据库中。

证书申请信息分为用户身份信息和证书相关信息两个部分。用户身份信息包括申请人姓名、身份证号、岗位或职务等。证书相关信息包括证书申请序列号、证书主体名称、证书拥有者的国家、组织和电子邮件等。证书申请信息还包括使用私钥的 PIN 密码，系统利用它对用户私钥进行保护，该密码以消息摘要的方式保存在数据库中。

(2) 身份核实

为了确保证书申请者身份的真实性，系统采用“面对面”确认方式进行身份验证：用户携带个人有效证件给 RA 管理员进行核实。RA 管理员根据用户给出的证书申请序列号从数据库中得到用户的身份信息和证书申请的相关信息，手工审核申请人真实身份和数据库中相关信息的一致性。系统通过修改数据库中证书的状态保存审核结果。例如，管理员将证书状态标志置为 1，代表已通过审核，等待 CA 生成证书；否则证书申请失败，RA 管理员删除该申请记录。

(3) 证书生成与颁发

CA 服务器从数据库中读取已通过审核的证书申请信息。针对每条证书申请信息，CA 利用 OpenSSL API 为申请人生成 RSA 密钥对，然后将私钥写入私钥库中存档；CA 利用 OpenSSL 提供的 API 为申请人生成并签发 X.509v3 格式的证书，将证书保存到数据库中，同时将证书的一个副本发布至 LDAP 目录服务器。

(4) 密钥和证书的发放

可以通过加密的 E-mail 方式将用户私钥以及证书序列号发送给用户，也可以通过带外方式实现密钥和证书的发放，例如将密钥复制到移动存储介质上，然后通知申请人领取。

(5) 证书查询

系统提供 Web 方式的证书查询系统，用户通过 Web 浏览器可以查询已经颁发的证书及其状态，单击相应的记录可以看到证书的详细信息。

(6) 证书更新

用户申请证书更新，提交申请证书时使用的 PIN(personal identification number)密码、身份证号码，以及原证书的序列号，若这些信息和数据库中保存的原始信息一致，CA 服务器就为该用户更新证书。证书更新过程和生成证书的过程相同，只是使用新生成的密钥对和证书来更新密钥表和证书表中的记录，并更新证书表中的证书有效日期。

(7) 密钥恢复

按如下步骤实现密钥(用户私钥)恢复。

① 用户通过 Web 页面向 RA 服务器提交密钥恢复申请,需要提供个人相关信息和 PIN 密码。

② RA 管理员手工审核用户真实身份,确认该请求来自用户本人。

③ 身份验证通过后,RA 服务器将请求提交给 CA 服务器。

④ CA 服务器从密钥表中获取该用户的密钥,通过加密的 E-Mail 或者带外方式提交给用户。

此外,系统还实现证书撤销、CRL 生成与发布等功能,这些均通过 OpenSSL 和 OpenLDAP 提供的 API 实现。

4.9.2 大型 PKI 系统设计案例

采用图 4.17 所示的体系结构。为了适应大型公共安全基础设施的需要,系统除了具有 CA、RA 和终端实体外,设立密钥管理中心(KMC),负责密钥的生成及存储管理。按照其存储密钥的状态,密钥库分为备用库、在用库和历史库三种。

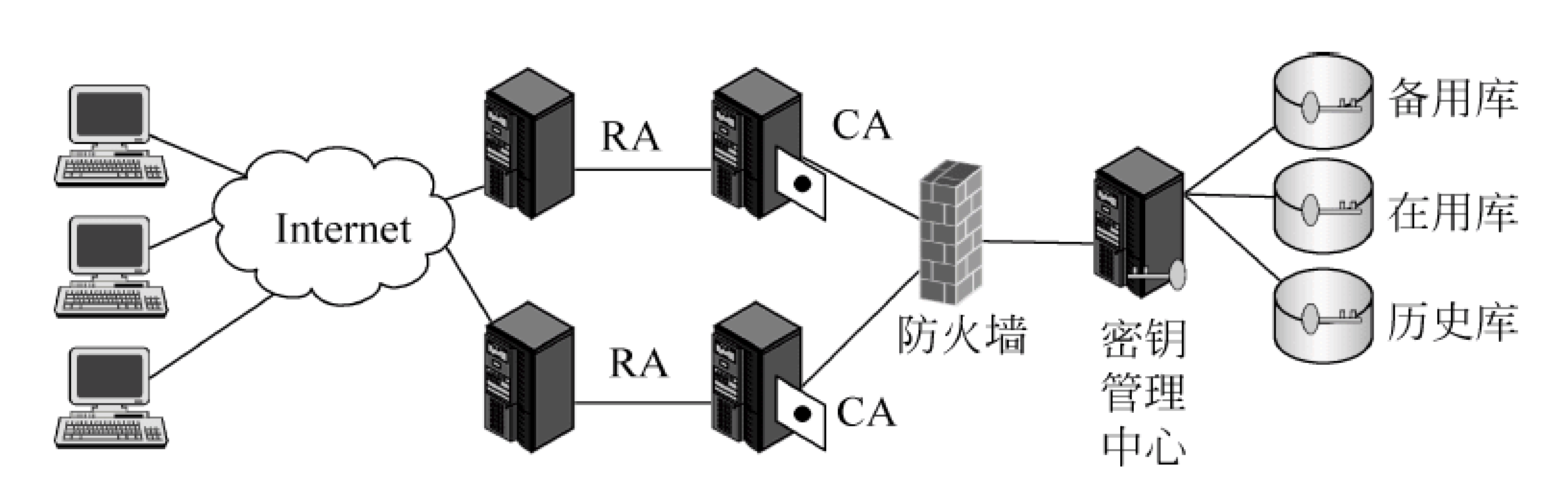


图 4.17 大型 PKI 体系结构示意图

用户的加密私钥托管在密钥管理中心。密钥管理中心提供对生命周期内加密证书密钥的全程管理功能,包括密钥生成、密钥存储、密钥分发、密钥备份、密钥更新、密钥撤销、密钥归档、密钥恢复以及密钥安全管理等。用户密钥由专门的密码机生成,并采用分割密钥存储方式保存于多个密钥分管者。

密钥管理中心不直接面向用户,而是通过 CA 与用户建立密钥托管关系。密钥管理中心与 CA 之间采用基于身份验证的安全通信协议,通信双方进行双向身份认证。密钥管理中心接收来自 CA 的请求,检查并确定请求合法后为 CA 提供相应的密钥管理服务,然后将结果返回给 CA。

CA 和 RA,以及 RA 和终端实体之间的通信按照 PKIX 模型中的 PKI 体系结构设计,和小型 PKI 系统具有类似功能。由于这种大型 PKI 系统提供公共的证书服务,很多用户通过 Internet 进行证书和密钥申请;同时,密钥管理中心可能服务于多个 CA,为它们提供密钥相关服务,因此,这种 PKI 系统对密钥的产生、托管、存储、分发以及恢复等提出更高的安

全性要求。

1. 密钥的产生

由密钥管理中心产生系统所需的各种密钥,包括如下内容。

① 根密钥:也称主密钥,是系统中最关键的密钥,可以是对称密钥,也可以是非对称密钥。

② 密钥保护密钥:用来保护其他密钥的密钥,其自身会受到根密钥的保护。

③ 终端密钥:也称用户密钥,属于某个终端设备或个人所有,可能是加密密钥或签名密钥,也可能是其他类型的密钥。

④ 会话密钥:也称临时密钥,这种密钥仅用于某一时段的通信保护或某一笔交易的保护,生存周期较短,其安全性要求相对较低。

系统的根密钥以及用户的终端密钥由硬件密码设备生成并保存在密码机中,采用密钥分割技术进行存储,由多个密钥分管者分别存放部分密钥分割信息。

2. 密钥的托管与存储

(1) 托管密钥的安全存储

系统根密钥及用户密钥由密钥管理中心采用分割密钥存储技术进行密钥托管,以保证密钥的安全性。将受托管的密钥分成 n 份后分别交给 n 个托管机构进行托管。这 n 个托管机构可以是 n 个密钥管理中心,也可以由某些 CA 和密钥管理中心组成。

保存被托管的密钥时,选定的密钥分管者分别使用各自的密码保护分管的密钥,分管的密钥存放在智能卡或智能密码钥匙中。智能卡或智能密码钥匙进行密钥备份,并安全存放。

在 KMC 受托管的终端密钥(用户密钥)由于数量庞大,影响面广,需要特殊的方案实现对这些密钥的保护。可以采用类似数字信封的多层密钥安全存储机制,用户密钥或密钥分量采用密钥加密密钥进行保护,然后再利用系统主密钥进行第二层加密保护。

(2) 终端密钥的安全存储

保存在用户端或终端设备中的用户密钥,其存储方式相对 KMC 的托管密钥来说安全性要求相对较低,因为单个终端密钥的泄露影响面相对较小。当然具体的安全要求还要看实际的应用环境。如果此终端密钥对于用户的安全系统非常重要,可以考虑采用针对关键密钥的存储方案。

终端密钥的存储技术一般有两种,一种是硬件保护,另一种是密码加密保护。选择终端密钥的硬件载体时,可以使用智能卡或智能密码钥匙一类的密码设备。这类密码设备的管理机制相对简单,一般通过密码来验证用户合法身份。

考虑到系统建设成本,也可以采用基于密码保护的加密方式。系统实施时,应按照 PKCS#8 规范,使用密码密钥保护 RSA 私钥。

3. KMC 与 CA 的安全通信

密钥管理中心进行用户密钥的分发时必须有安全可靠的通信协议作为支撑。因此,为

KMC 设计专门的安全密钥分发协议：KMC 在收到来自 CA 的业务请求后，首先检查请求的合法性与正确性，然后根据 CA 的请求内容进行相应的处理，并将结果返回给 CA。KMC 为 CA 提供服务的完整过程，包括请求、响应、回执以及异常情况的处理等。

其中，CA 的服务请求包括如下内容：

- 协议版本。
- 服务请求标识符。
- CA 证书标识符。
- 扩展的请求信息。
- 请求信息的签名。

KMC 的响应包括如下内容：

- 协议版本。
- 响应标识符。
- KMC 证书标识符。
- 响应信息的签名。

密钥管理系统在运行过程中会涉及多个功能模块之间的相互调用，以及各种管理员的操作，对这些调用和操作需要以日志的形式进行记录，以用于系统错误分析、风险分析和安全审计等工作。这些日志信息包括调用请求的接收时间、请求者的网络地址、身份、请求的内容、请求处理过程和处理结果等。

4. 密钥恢复

两种情况下需要进行密钥恢复：用户对自己私钥的恢复和司法取证的密钥恢复。用户对自己私钥的恢复过程与密钥的下载过程类似，只需要用户证明自己的身份，即可分别从不同密钥托管者那里获得密钥分量，经过计算后恢复自己的私钥。

司法取证的密钥恢复过程在算法实现上与用户密钥下载过程相同，区别是司法取证的密钥恢复过程需要通过行政管理手段保证其合法性和安全性。政府有关部门在需要进行司法密钥恢复时，需要同时向 CA 和 KMC 提供司法部门的书面证明文件，CA 和 KMC 分别在超级管理员授权的情况下签发一个具有一定时效的司法密钥恢复卡。在向 KMC 和 CA 分别证明自己的身份后可以获得密钥分量再恢复相应的私钥数据。所恢复出的密钥必须在使用结束后，在 KMC 的管理员的监督下进行销毁，以防止永久监听情况的发生。

4.9.3 PKI 应用简介

PKI 是一种提供非对称密码体制中密钥的封装（公钥封装为证书）、分发和管理的基础设施，因此，理论上讲，凡是使用非对称密码的所有协议和应用都可以使用 PKI 技术。应用 PKI 可提供身份认证、数据机密性服务、不可否认性服务和数据完整性服务等，但 PKI 本身

并不提供面向终端用户或应用系统的加密或签名服务,这些服务通过基于 PKI 的应用协议或应用系统实现(如图 4.13 所示),如安全 Web 应用、安全电子邮件、安全电子商务和虚拟专用网等。

如图 4.13 所示,基于 PKI 的应用包括两类,一类是使用 PKI 的安全协议,如 IPSec、SSL/TLS 和 S/MIME 等,在这些安全协议的基础上构建各种安全应用系统或服务;另一类是直接使用 PKI 进行加密或签名的应用层安全服务,如电子签章、数字信封和单点登录等。

1. SSL/TLS 与安全 Web 应用

由于 TCP/IP 协议没有提供安全能力,HTTP 等应用层协议提供的服务在不安全的网络上完成,应用层的通信过程可能被窃听、篡改和欺骗。这些安全隐患限制了早期的 Internet 只能用于传输一些公开性和安全性要求不高的信息,阻碍了 Web 应用的进一步发展。

安全套接字(secure sockets layer,SSL)协议是位于网络传输层和应用层之间的安全通信层协议,SSL 协议中的两个实体进行通信之前,首先要建立 SSL 连接,进行身份认证,从而防止未授权访问。

SSL 最初由 Netscape 公司开发,该协议向基于 TCP/IP 的客户及服务器应用程序提供客户端和服务器的鉴别、消息机密性及完整性等安全服务。Internet 工程任务组 IETF 以 SSL 为基础制定了传输层安全(transport layer security,TLS)协议标准。TLS 与 SSL V3.1 十分类似,两者基本可互相兼容。

利用 PKI 技术,SSL 协议允许在客户机和服务器之间进行数据交换前通过交换 SSL 初始握手信息来实现有关安全特性的审查,并在数据交换过程中采用 DES、MD5 等加密技术进行加密通信,以保证信息的机密性和完整性。此外,SSL 还利用数字证书保证通信安全,通信双方通过验证对方的数字证书来确认对方的身份,并在此基础上提供通信内容的机密性服务。

结合 SSL 协议和数字证书,PKI 技术可以满足 Web 交易的安全需求,因此,SSL/TLS 已被广泛应用于基于 Web 的安全应用系统中,如网上银行、网上购物和安全电子商务等。基于 SSL/TLS 的 WWW 协议称为 HTTPS(HTTP over SSL)。HTTPS 中的浏览器和 Web 服务器之间需要交换数字证书,进行身份鉴别,从而防止 Web 应用的通信实体被冒充。

SSL 协议也是国际上最早应用于电子商务的一种网络安全协议,至今仍然有许多网上商店在使用 SSL 协议。SSL 协议在点对点的网上银行业务中也经常使用。在电子商务交易过程中,由于有银行参与,按照 SSL 协议,客户的购买信息首先发往商家,商家再将信息转发给银行,银行验证客户信息的合法性后,通知商家付款成功,商家再通知客户购买成功,并将商品寄送客户。

2. 安全电子邮件

电子邮件凭借其易用、低成本和高效已经成为互联网时代的一种标准信息交换工具。随着 Internet 的持续增长，商业机构或政府机构都开始用电子邮件交换一些秘密的或是有商业价值的信息，这就引出了一些安全方面的问题，包括：

- 消息和附件可以在不为通信双方所知的情况下被阅读、截取或篡改。
- 发信者的身份可能被人伪造。

前一个是安全问题，后一个是信任问题，正是由于安全和信任的缺乏，尽管电子邮件本身具有众多优点，仍然使得许多企业、机构不使用电子邮件交换关键信息。

和安全 Web 应用类似，电子邮件的安全需求也包括身份认证、机密性、完整性和不可否认性，这些均可利用 PKI 技术获取。具体地说，利用数字证书和私钥，用户可以对他所发的邮件进行数字签名，这样就可以获得认证、消息完整性和不可否认性服务，如果证书是由其所属公司或某一可信第三方颁发的，收到邮件的人就可以信任该邮件的来源，无论他是否认识发邮件的人；另一方面，在政策和法律允许的情况下，用加密的方法就可以保障信息的机密性。

PGP 加密已经在电子邮件通信中得到了一定范围内的应用，它也是一种公钥加密体制，但它不是基于 PKI 的，没有权威机构进行公钥认证，也不使用 X.509 证书。因此 PGP 不适合在提供公共服务的电子邮件服务器和应用系统中使用，使用 PGP 的双方需要自己进行安全沟通和公钥交换。

安全电子邮件协议 S/MIME(the secure multipurpose internet mail extension)是一个允许发送加密和具有签名邮件的协议。该协议的实现需要依赖于 PKI 技术。基于 PKI 的安全电子邮件则具有普遍意义，因为 PKI 的用户群可以是开放的。

S/MIME 支持邮件的签名和加密。基于 MIME 和 PKI 标准，S/MIME 为电子消息应用程序提供认证、完整性保护及数据机密性等安全服务。传统的邮件用户代理可以使用 S/MIME 来加密发送邮件及解密接收邮件。S/MIME 并不仅限于邮件的使用，它也能应用于任何可以传送 MIME 数据的传输机制，例如 HTTP。同样，S/MIME 利用 MIME 的面向对象特征允许在混合传输系统中交换安全消息。

S/MIME 最初版本 V1 来源于私有的商业社团 RSA 数据安全公司。S/MIME V2 版本已经广泛地使用在安全电子邮件上，但是它并不是 IETF 的标准，因为它需要使用 RSA 的密钥交换，这就受限于美国 RSA 数据安全公司的专利。目前 S/MIME 最新版本 S/MIME V3 已成为 IETF 标准的一部分。

同 PGP 一样，S/MIME 也利用单向散列算法和公钥与私钥的加密体系。但它的认证机制依赖于层次结构的证书认证机构及 PKI，并且使用数字证书进行身份鉴别，因此比 PGP 具有更好的通用性，并能满足更高的安全性需求。S/MIME 将信件内容加密签名后作为特殊的附件传送，它的证书格式采用 X.509，但与一般浏览器网上使用的 SSL 证书有一

定差异。

国内众多的认证机构基本都提供“安全电子邮件证书”的服务，其技术对应的就是 S/MIME 技术，平台使用的基本上是美国 Versign 的。

3. IPSec 与虚拟专用网

虚拟专用网(VPN)是一种架构在公用通信基础设施之上的专用数据通信网络，利用 IPsec 等网络层安全协议和基于 PKI 的加密与签名技术来获得通信的私有性。同租用线路等专用网络相比，VPN 既节省开销又易于安装和使用，适合企业架构 Intranet 和进行远程办公场地的互联。

通常，企业在架构 VPN 时都会利用防火墙和访问控制技术来提高 VPN 的安全性，这只解决了很少一部分问题，而一个现代 VPN 所需要的安全保障，如身份认证、机密性、完整性和不可否认性等，都需要采用更完善的安全技术来实现。在实现上，VPN 的基本思想是采用秘密通信通道，用加密的方法来保障通信的安全性和私用性。

VPN 的实现协议一般有 PPTP、L2TP 和 IPSec 三种。其中，PPTP(point to point tunneling protocol)是点对点的协议，基于拨号使用的 PPP 协议使用 PAP 或 CHAP 之类的加密算法，或者使用 Microsoft 的点对点加密算法 MPPE。而 L2TP(layer2 tunneling protocol)是 L2FP(layer 2 forwarding protocol)和 PPTP 的结合，依赖 PPP 协议建立拨号连接，加密的方法也类似于 PPTP，但这是一个两层的协议，可以支持非 IP 协议数据包的传输，如 ATM 或 X.25，因此也可以说 L2TP 是 PPTP 在实际应用环境中的推广。PPTP 和 L2TP 不支持 PKI 技术和数字证书，缺乏数字证书的 VPN 对身份认证、完整性和不可否认性的支持相对较差。因此，这两种 VPN 方式对现代安全需求的支持都不够完善，应用范围也不够广泛。

IPSec 协议由 IETF 工作组开发，最初一组有关 IPSec 的标准在 1995 年制定，它是一个应用广泛、开放的 Internet 标准协议，目前已经成为最流行的 VPN 解决方案。IPSec 包括 AH 和 ESP。AH 验证头提供数据源身份认证、数据完整性保护、重放攻击保护功能；ESP 安全负载封装提供数据保密、数据源身份认证、数据完整性、重放攻击保护功能。IPSec 可以为路由器之间、防火墙之间或者路由器和防火墙之间以及端用户之间提供经过加密和认证的通信。

IPSec 的身份验证可以基于 PKI 技术实现，采用 X.509 证书提供身份鉴别，因此可提供比 PPTP 和 L2TP 更广泛的安全应用。由于 IPSec 是 IP 层的安全协议，因此很容易在全世界范围内形成一种规范，具有良好的通用性，并且能够很好地支持面向未来的协议—— IPv6。

IPSec 还是一个发展中的协议，随着成熟的 PKI 技术的应用，公钥密码技术越来越多地被嵌入到 IPSec 中，相信在未来几年内，该协议在 VPN 的应用中会更加广泛。

4. 其他应用层安全服务

(1) 数字信封

数字信封的功能类似于普通信封。普通信封在法律的约束下保证只有收信人才能阅读信的内容；数字信封则采用密码技术保证了只有规定的接收人才能阅读信息的内容。

数字信封中采用了单钥密码体制和公钥密码体制。信息发送者首先利用随机产生的对称密码加密信息，再利用接收方的 PKI 公钥证书加密对称密码，被公钥加密后的对称密码称为数字信封。在传递信息时，信息接收方要解密信息时，必须先用自己的私钥解密数字信封，得到对称密码，才能利用对称密码解密所得到的信息。这样就保证了数据传输的真实性和完整性。

(2) 数字印章与数字水印

数字印章的原理和数字签名相同，利用签名者的私钥对数据的消息摘要进行签名，接收者利用签名者的 PKI 公钥证书对签名进行验证。数字印章是对数字签名的图形化，首先为应用系统设计一个签名图章，然后通过程序接口嵌入某种格式的文档中(如 PDF、WORD 等)，在嵌入印章的同时，对该文档进行数字签名，然后将公钥携带在文档中发送给对方进行印章或签名的验证。

数字水印是一种信息隐藏技术，它将数字信号，如图像、文字、符号、数字等一切可以作为标记、标识的信息与原始数据(如图像、音频和视频数据)紧密结合并隐藏其中，并可以经历一些不破坏源数据价值的操作而能保存下来。电子印章系统采用易碎水印来保护印章图像，当印章图像被更改后，即使是一个像素，都会破坏水印本身，从而达到验证保护的目的。

此外，数字水印还可被广泛应用于信息防伪、版权保护、篡改提示和信息隐蔽等多个领域。

(3) 单点登录

单点登录(single sign-on，SSO)是一种方便用户访问多个系统的技术，用户只需在登录时进行一次注册，就可以访问多个系统，不必重复输入用户名和密码来确定身份。单点登录的实质就是安全上下文(security context)或凭证(credential)在多个应用系统之间的传递或共享。当用户登录系统时，客户端软件根据用户的凭证(例如用户名和密码)为用户建立一个安全上下文，安全上下文包含用于验证用户的安全信息，系统用这个安全上下文和安全策略来判断用户是否具有访问系统资源的权限。

CA 提供了统一认证的功能，为了在单点登录中实现统一的认证和授权的功能，需要引入授权管理中(privilege management Infrastructure，PMI)。PMI 的目标是向用户和应用程序提供授权管理服务，提供用户身份到应用授权的映射功能，提供与实际应用处理模式相对应的、与具体应用系统开发和管理无关的授权和访问控制机制，简化具体应用系统的开发与维护。PMI 是属性证书(attribute certificate)、属性权威(attribute authority)和属性证书库等部件的集合体，用来实现权限和证书的产生、管理、存储、分发和撤销等功能。SSO 和

PMI 需要依靠 PKI 为其提供身份认证服务。

本章实验

1. 利用 Windows 2000 Server 中的 CA 进行证书的申请和颁发，并将签名后的证书安装在浏览器中，作为可信的证书。

2. 在 Tomcat 中生成密钥并生成 CA 签名的证书。

思考题

1. 对于客户端来说，什么样的 CA 是可信的？

2. 常见的 CA 的信任模型有哪几种？它们分别有什么特点？这些信任模型的实施方法是怎样的？

3. 一个内部使用的 CA 系统和一个面向公共服务的 PKI 系统在设计上的主要区别是什么？它们分别着重考虑哪些安全问题？

4. 假设客户端拿到一个服务器端的证书，为了验证证书的有效性，客户端应该分别检查证书的哪些信息？客户端如何验证证书上颁发者签名的有效性？

第二部分

TCP/IP 网络安全协议

第5章

网络层安全协议

5.1 IPSec概述

Internet 的网络层采用 IP 协议，但是传统 IPv4 没有提供安全服务：缺乏对通信双方身份真实性的鉴别能力，而且没有提供传输数据的完整性和机密性保护。因此，Internet 的网络层面临业务流监听、IP 地址欺骗、信息泄露和数据项篡改等多种安全威胁。

IPSec(Internet protocol security)即 Internet 安全协议，是 Internet 工作组 IETF 提出的保护 IP 报文安全通信的一系列规范，它提供私有信息通过公用网的安全保障。IPSec 是一簇协议，用于在 IP 层提供机密性、数据源鉴别和完整性保护。IPSec 本身并不规定协议使用的加密、鉴别和完整性保护算法，也不限制用户使用以上提供的哪一种或几种服务，这些依赖具体 IPSec 的协议实现以及用户对安全服务及其参数进行协商的结果，因此，可以认为 IPSec 是一个协议框架。

由于 IPSec 在 TCP/IP 协议的核心层——IP 层实现，因此可以有效地保护各种上层协议，并为各种安全服务提供一个统一的平台。IPSec 是目前虚拟专用网中使用最广泛的一种协议，也是下一代 Internet 所采用的网络安全协议。IPSec 是随着 IPv6 的制定而产生的，它在 IPv6 中是必须支持的协议。鉴于 IPv4 的应用仍然很广泛，所以在 IPSec 的制定中也增加了对 IPv4 的支持。

使用 IPSec 时，用户可以有选择地使用其中的一项或多项功能，从而得到期望的安全服务。IPSec 可保障主机之间、网络安全网关(如路由器或防火墙)之间或主机与安全网关之间 IP 报文的安全。使用 IPSec 后，用户的原始 IP 报文进行加密或完整性保护等处理后，生成新的 IP 报文在网络上传输。

5.2 IPSec 体系结构

如图 5.1 所示，IPSec 协议框架主要包括验证头(AH)、封装安全载荷(encapsulating security payload，ESP)、密钥管理、安全策略等。其中，AH 和 ESP 是 IPSec 的安全通信子协议，安全通信使用的密钥可以通过密钥交换协议得到。

1. 验证头协议

AH 提供无连接的完整性、数据源认证和抗重放保护服务，但 AH 不提供消息的机密性服务。AH 的主要作用是为 IP 报文提供密码认证，以确保数据完整、来源可靠，并且没有被重放。其中，数据完整性保护通过 HMAC 实现，抗重放攻击通过 AH 报文的序列号实现。数据完整性算法、数据源认证算法取决于 IPSec 协议的具体实现，并且可以由通信双方协商后确定。

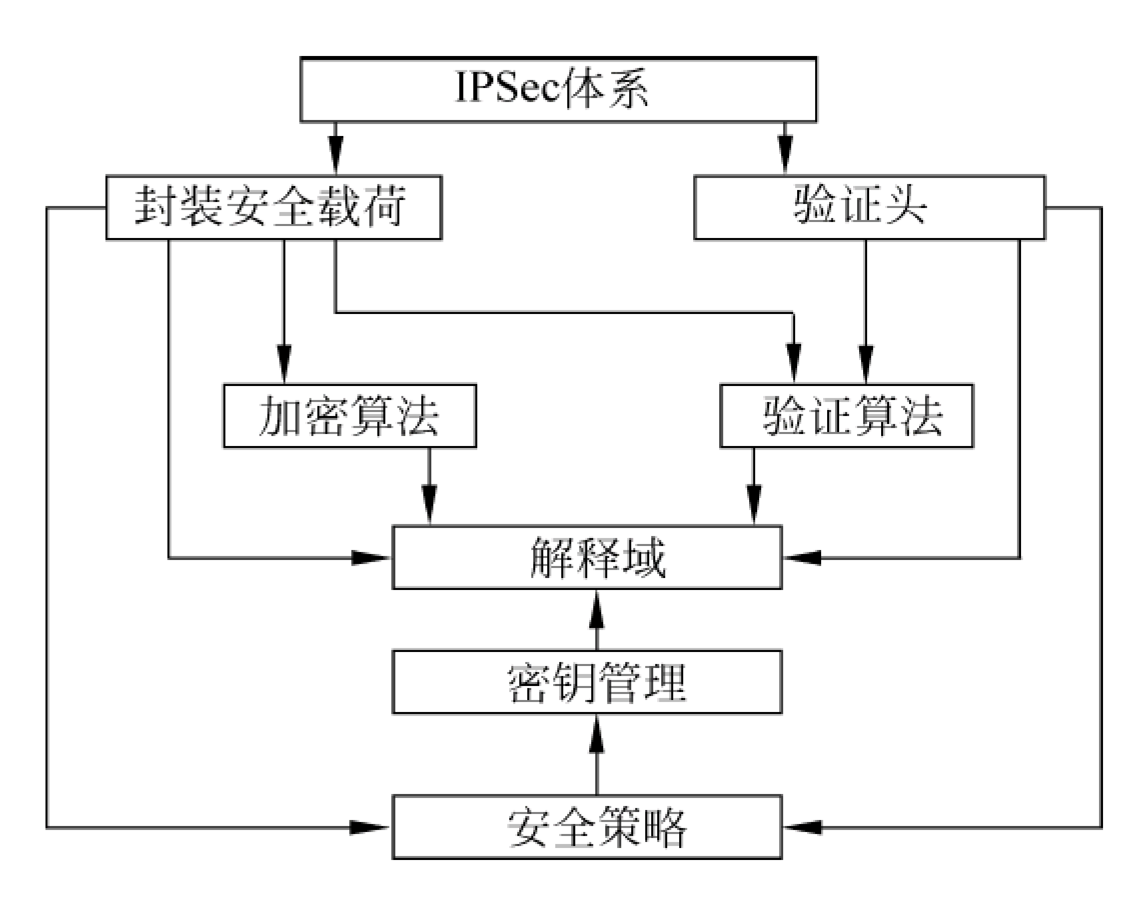


图 5.1 IPSec 框架的体系结构

2. 封装安全载荷协议

ESP 为 IP 提供机密性、数据源验证、抗重放保护以及数据完整性保护等安全服务。其中，数据机密性是 ESP 的基本功能，而数据源身份认证、数据完整性检验以及抗重放保护等功能是可选的，这由通信双方协商决定。因此，ESP 可以同时使用加密算法和验证算法，也可以单独使用加密算法或验证算法。ESP 使用对称加密算法提供机密性服务，使用的密码算法取决于 IPSec 的实现，并且可以由通信双方协商后确定。

ESP 可以和 AH 联合使用，也可以单独使用。

3. 安全策略

IPSec 允许用户控制安全服务的粒度，这通过安全策略(security policy，SP)实现。简言之，IPSec 安全策略规定了一个通信实体使用 IPSec 进行安全通信中的目标网络或地址、使用的安全参数等一系列要素。安全策略保存在通信实体的本地安全策略数据库(security policy database，SPD)中。

例如，一个组织的安全策略可能规定来自特定子网的数据流使用 AH 和 ESP 保护，并且使用 DES(数据加密标准)算法加密 IP 报文，使用 MD5 算法验证数据的完整性。另一方面，安全策略可能规定来自另一个站点的数据流只使用 ESP 进行机密性保护，并且使用高级加密标准(advanced encryption standard，AES)进行加密。

4. 安全协定(security association,SA)

SA 是保证 IPSec 通信双方协调工作的基础,是两个使用 IPSec 进行安全通信的通信实体经协商建立的一种协定,它决定了用来保护 IP 报文安全的一系列要素,如使用的安全通信协议(AH 或 ESP 协议)、转码方式、密钥、密钥算法及密钥的有效时间等。SA 中的这些要素由通信双方协商产生,并保存在安全协定数据库(security association database,SAD)中。SA 可以手工创建,也可以动态生成,动态建立 SA 使用密钥管理协议实现。

安全策略包括一个指向安全协定的指针。因此,可以认为安全协定是和安全策略相关联的一部分。安全策略和安全协定的内容均由通信实体协商确定,并保存在各自的数据库中(SPD 和 SAD),进行安全通信(使用 AH、ESP)时,IPSec 根据 SPD 和 SAD 中对应的参数值完成鉴别和加密过程。

5. 密钥管理

IPSec 在数据包验证和加密过程中需要使用各种密钥,这些密钥可以由密钥管理组件进行分发和管理,通过密钥管理协议来实现。密钥管理协议进行密钥协商的结果(例如密钥及其长度等)被保存在 SA 中。Internet 密钥交换协议(internet key exchange,IKE)是 IPSec 默认的安全密钥协商方法。IKE 通过一系列报文交换,为两个实体(如网络终端或网关)进行安全通信生成会话密钥。IKE 建立在 Internet 安全协定和密钥管理协议(internet security association and key management protocol,ISAKMP)定义的一个框架之上,是 IPSec 目前正式确定的密钥交换协议。IKE 为 IPSec 的 AH 和 ESP 协议提供密钥交换管理和安全协定管理,同时也为 ISAKMP 提供密钥管理和安全管理。IKE 具有两种密钥管理协议,即 Oakley 和安全密钥交换机制(secure key exchange mechanism,SKEME)的一部分功能,并综合了 Oakley 和 SKEME 的密钥交换方案,形成了独有的加密密钥生成方法。

6. 解释域(domain of interpret,DOI)

给出各个组件彼此相关部分的标识符及操作参数。

5.3 Ipsec 的操作模式

IPSec 有两种操作模式(operation mode),即隧道模式和传输模式。IPSec 隧道模式的特点是数据包目的地不是 IPSec 的安全终点。通常情况下,只要 IPSec 通信双方有一方是安全网关或路由器,就必须使用隧道模式。传输模式下,IP 报文的源和目标地址是 IPSec 的安全终点。通常情况下,传输模式用于两台主机之间的端到端安全通信。

两种操作模式的示意图分别如图 5.2 和图 5.3 所示。

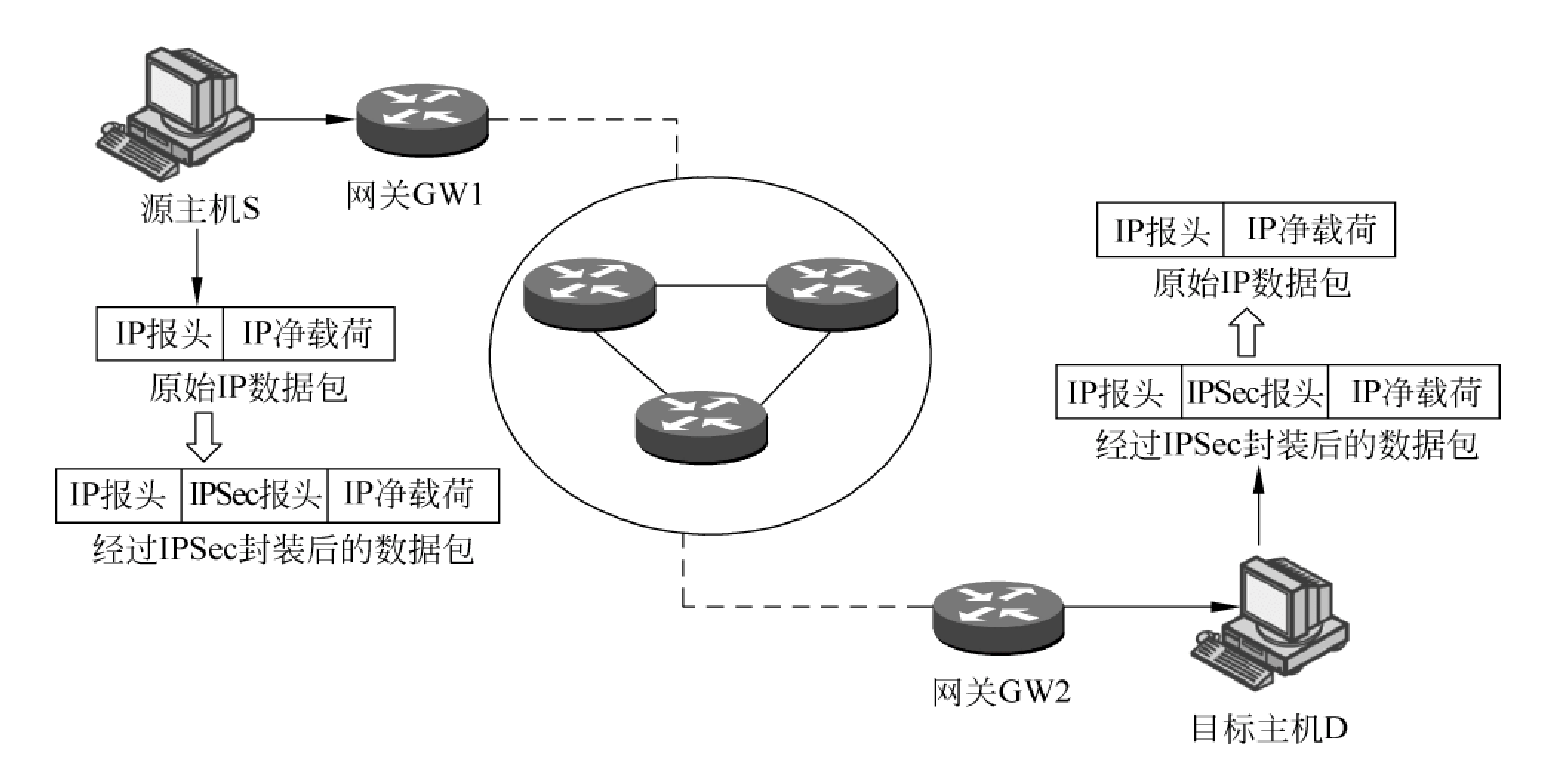


图 5.2　IPSec 传输模式

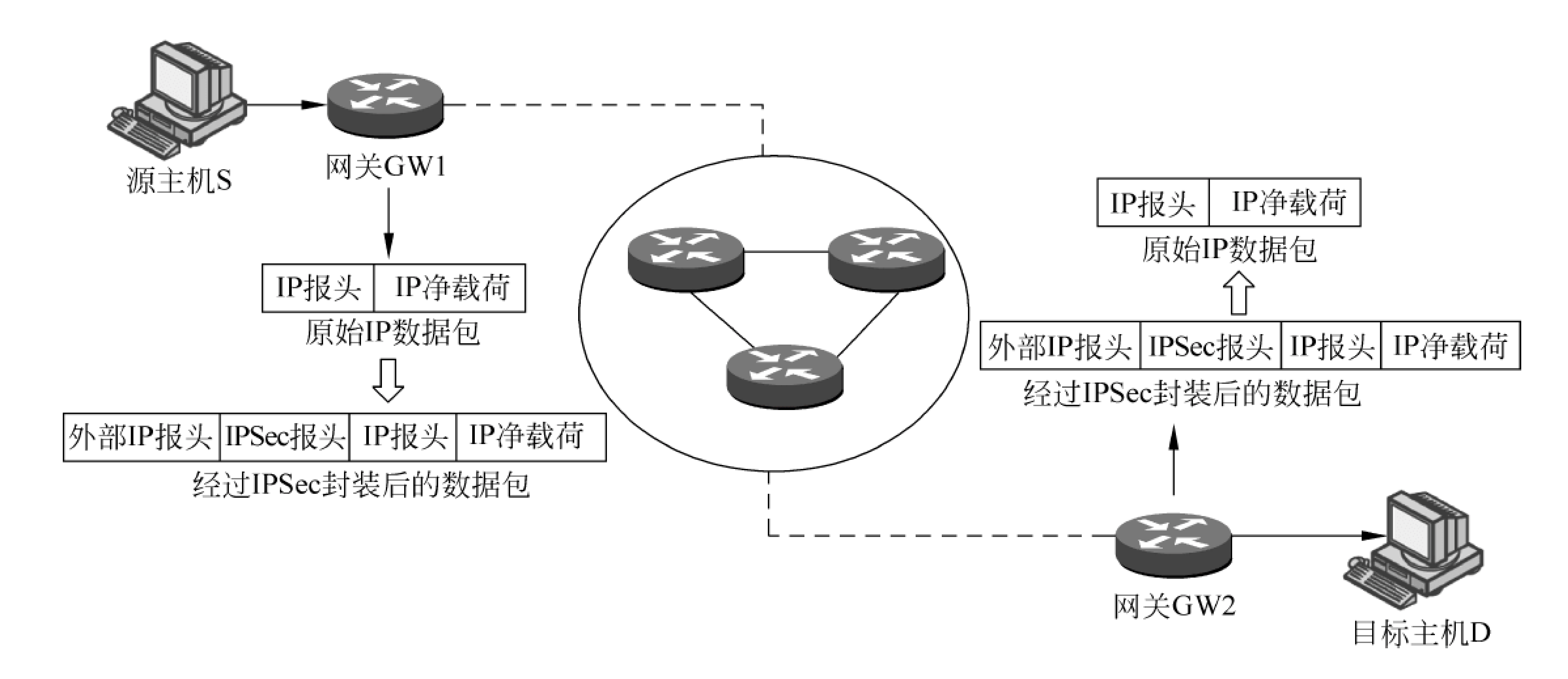


图 5.3　IPSec 隧道模式

传输模式中，IPSec 对等实体位于两台主机上：一台主机向另一台主机发送 IP 报文，两台主机上均启动 IPSec 协议，并按照事先协商的安全策略和安全协定对 IP 报文进行安全通信保护。在源主机上，IPSec 安全通信协议 AH 和/或 ESP 的头部被插入到原 IP 报文的头部和其净载荷(上层协议)之间，该数据包到达目标主机后，目标主机的 IPSec 协议栈进行相应的安全操作(如进行数据源鉴别、完整性验证和解密等)后，将数据包交由上层协议处理。

隧道模式中，IPSec 对等实体位于两个安全网关之间(或位于主机与安全网关之间)：一台主机向另一台主机发送一个明文 IP 报文，通信目标端不是 IPSec 的协议终点(目标主机不启动 IPSec)，该数据包到达本地网关后，由本地网关和目标地址的网关之间建立 IPSec 隧

道。本地网关的 IPSec 协议栈对该 IP 报文进行重新封装,原始 IP 报文经过 AH 和/或 ESP 封装保护后,IPSec 产生一个新的 IP 报头,报头的源地址为本地网关,目标地址为远程的对端网关(目标主机所在网关)。该数据包到达目标地址的网关后,由远程网关进行 IPSec 安全检查后,将数据包还原为原始明文 IP 报文,并转发给目标主机。图 5.3 中的隧道在两个安全网关之间产生,也可以在主机和目标网络的网关之间产生,此时数据包的封装及处理方式相同。

如图 5.4 所示,在实际部署时,IPSec 隧道模式和传输模式可以共同存在,网络上的主机和网络设备根据需要配置为不同的工作模式,两种模式也可以嵌套使用,共同对网络上的 IP 通信进行保护。

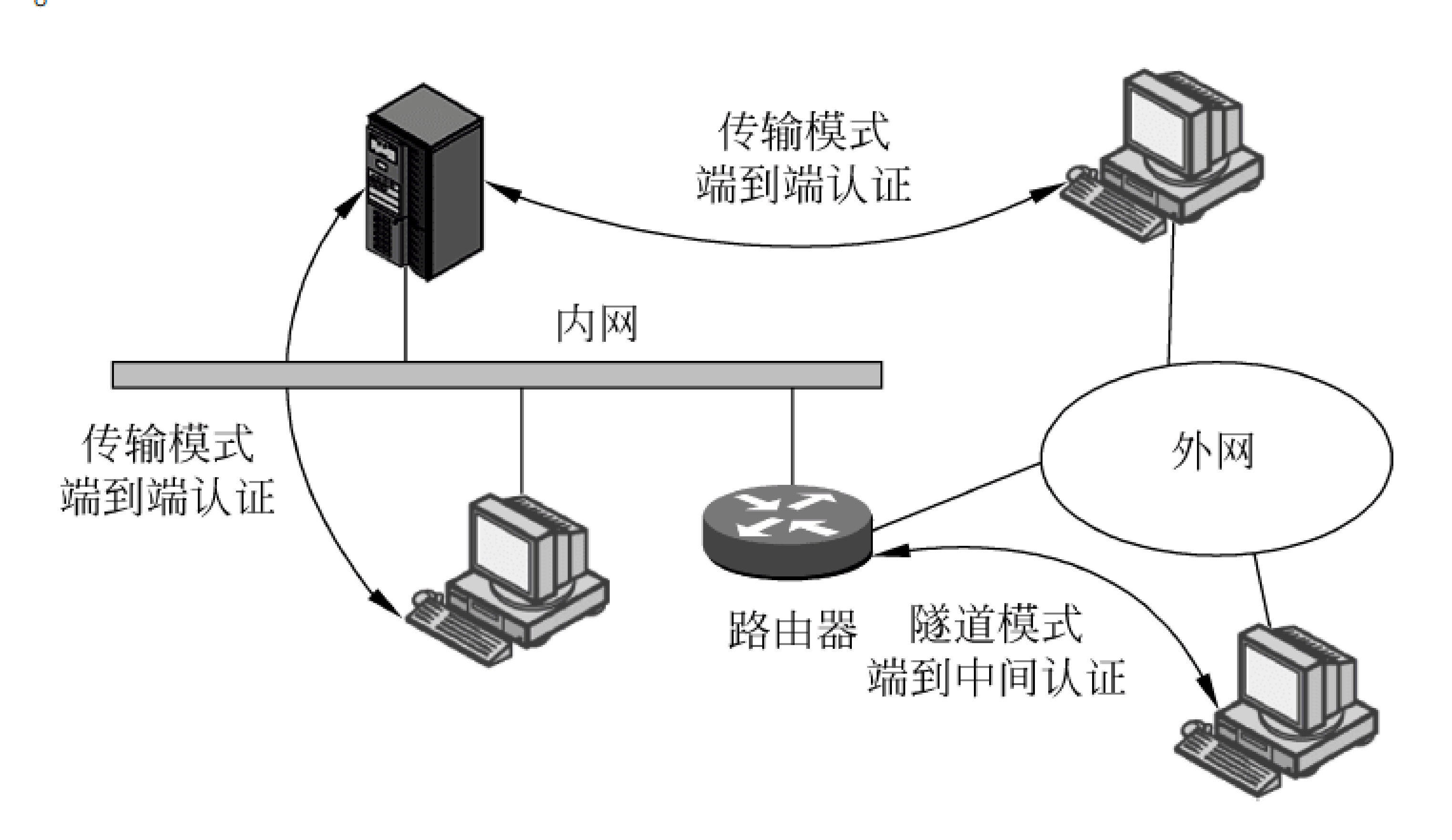


图 5.4　两种模式在网络中的部署

5.4　安全策略与安全协议

对于 IPSec 数据流处理而言,有两个必要的数据库:安全策略数据库(SPD)和安全协定数据库(SAD)。SPD 指定了用于到达或者源自特定主机或网络的数据流的策略。SAD 则包含安全通信协议需要的各种参数。

1. 安全策略和安全策略数据库

如图 5.5 所示,IPSec 安全策略包括需要保护的通信双方的一系列参数。IPSec 协议要求在所有通信流处理的过程中都必须查询 SPD。SPD 中包含一个策略条目的有序列表,通过使用一个或者多个选择符(如图 5.5 中所示的目的地址、源地址、传输层协议、系统名和上层协议等)来确定每个条目,这些选择符是从网络层和传输层的数据包头里提取出来的。SPD 中的 SA 指针用来确定该策略对应的安全协定。当有数据包外出时,根据这些选择符对 SPD 进行索引,从而决定采用何种安全策略和安全协定。

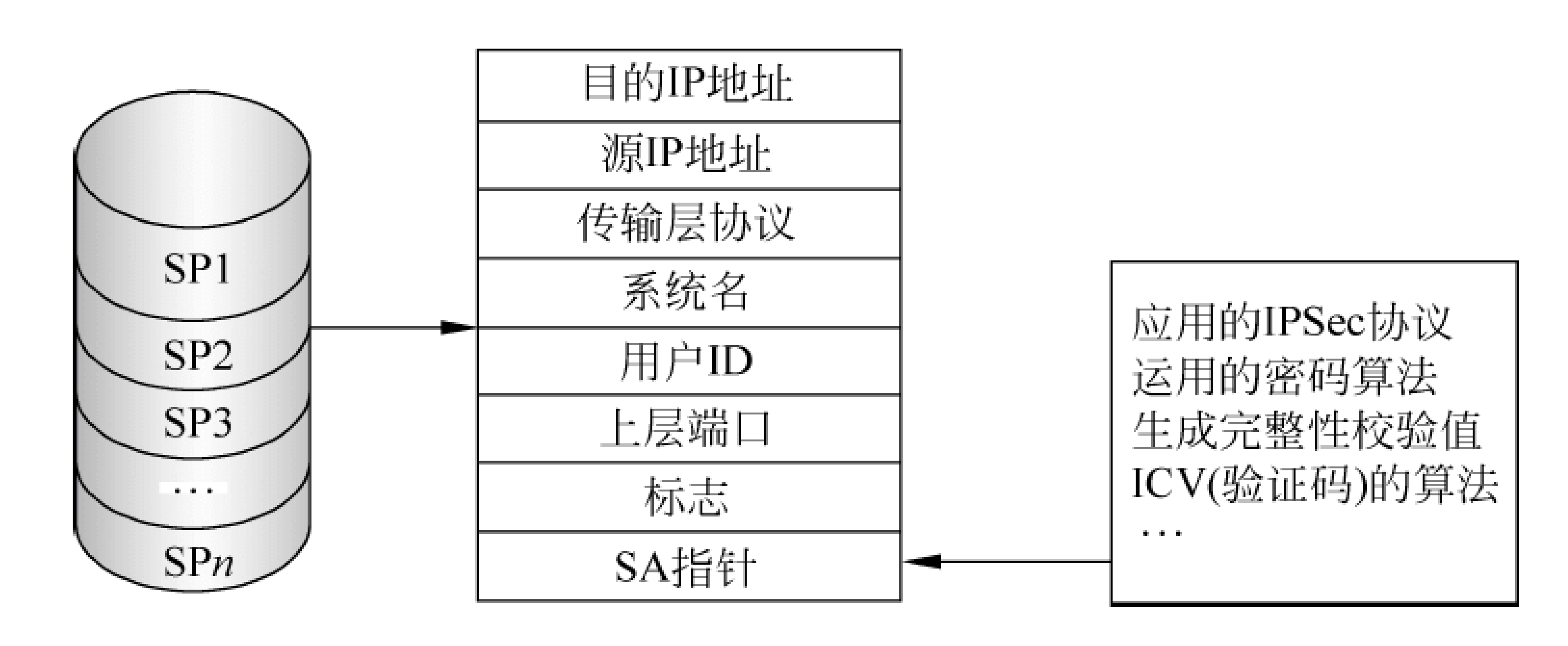


图 5.5　IPSec 安全策略 SP

IPSec 当前允许的选择符包括如下几个部分。

- 源 IP 地址。源 IP 地址可以是一个 32 位的 IPv4 地址,也可以是一个 128 位的 IPv6 地址。该地址可以是通配符、地址范围、网络前缀或是指定主机的地址。对源自一个主机的所有 IP 报文来说,假如为它们采取的安全通信策略相同,则使用通配符作为源地址。网络前缀和地址范围用于安全网关,以便为隐藏在它后面的主机提供安全保护,以及用来构建 VPN。在一个主机安全要求已经明确的前提下,要么在一个多宿主(多穴)主机上使用一个特定的地址,要么在网关上使用。源 IP 地址从 AH、ESP 或者 IP 报头的源 IP 地址域中得到。
- 目的 IP 地址。目标 IP 地址也可以是一个 32 位的 IPv4 地址,或 128 位的 IPv6 地址。同样,该地址也可以是通配符、地址范围、网络前缀或指定主机。前三个都用于隐藏在安全网关后的主机。目的 IP 地址从 AH、ESP 或者 IP 头(若没有对数据包应用 IPSec 的话)的目的 IP 地址域中得到。对于经过隧道处理的 IP 报文,用作选择符的目标地址字段有别于用于查找 SA 的目标地址。在这种情况下,只要数据包以隧道方式传输,外部 IP 报头的目的 IP 地址便可与内部 IP 报头的不一样。但是,目的网关中的策略是根据真正的目标地址设定的,而且最终要使用这一地址对 SPD 数据库进行索引。
- 系统名。系统名字段用于标识与一名有效用户或者系统名称关联的策略。它可以是一个 DNS 名、X.500 名或者在 IPSec 中定义的其他名字类型。只有在 IKE 协商期间(而非包处理期间),名字字段才能作为一个选择符使用。在包处理期间,这一字段不能作为选择符使用。这是由于目前无法把一个 IP 地址和一个名字结合在一起。
- 协议。协议字段指定了传输层协议,该字段的值可以从 IPv4 或 IPv6 报头的“下一个头”域中得到。许多情况下,只要使用了 ESP,传输层数据便无法被访问,因为它作为 IP 有效载荷被加密了。这种情形下,需要使用通配符。
- 上层端口。在进行面向会话的密钥交换时,上层端口代表源和目标端口,真正应用协议的便是这些端口。如果端口不能访问,便需要使用通配符。SPD 中的每一个条

目都包含一个或者多个选择符和一个标志，该标志用于表明与条目中的选择符匹配的数据包是否应该丢弃、进行 IPSec 处理或者绕过 IPSec 处理。如果应该对数据包进行 IPSec 处理，则条目中必须包含一个指向 SA 内容的指针，其中详细说明了应用于匹配该策略条目的数据包的 IPSec 安全子协议（AH 或 ESP）、操作模式（隧道或传输模式）和加密验证算法等。选择符与数据通信流相匹配的第一个条目将被应用到该通信中。如果没有发现匹配的条目，通信数据包将被丢弃。

2. 安全协议和安全协定数据库

SAD 中包含现行的 SA 条目，每个 SA 又包含一个安全参数索引（security parameters index，SPI），一个源或目的 IP 地址和一个 IPSec 安全通信协议（AH/ESP）的三元组索引，即＜SPI，dst/src，protocol＞。该索引唯一标识一个 SA，并且作为 AH 和 ESP 协议报文头部的一个字段存在。此外，一个 SAD 条目还包含下面的域：

- 序列号（sequence number）。序列号是一个 32 位的整数，用于生成 AH 或 ESP 协议报头中的序列号域，在数据包"外出"处理期间使用。它同时属于 AH 及 ESP 报头的一部分。每次用 SA 来保护一个数据包，序列号的值便会递增 1。通信的目标主机利用这个字段来检测"重放"攻击。SA 刚刚建立时，该字段的值设为 0。通常，在这个字段的值溢出之前，SA 会重新进行协商。
- 序列号溢出。这是一个标志，用于外出包处理，表示是否对序列号计数器的溢出进行审核；对于特定的 SA，是否阻塞额外通信流的传输。
- 抗重放窗口。该字段在数据包的"进入"处理期间使用。它使用一个 32 位计数器和位图确定一个输入的 AH 或者 ESP 报文是否是一个重放包。
- AH 认证密码算法和所需要的密钥。指定 AH 安全子协议中密码算法、密钥等重要安全参数。
- ESP 认证密码算法和所需要的密钥。指定 ESP 安全子协议中密码算法、密钥等重要安全参数。
- ESP 加密算法，密钥，初始化向量（initialization vector，IV）和 IV 模式。其中，初始化向量 IV 是一个随机数，每次加密时随机产生。IV 以某种形式与原密钥相组合，作为该次加密的加密密钥，用来解决密钥重用的问题。
- IPSec 操作模式。IPSec 协议可同时用于隧道模式及传输模式。依据这个字段的值，载荷的处理方式也会有所区别。可将该字段设为隧道模式、传输模式或者一个通配符。若将这个字段设为通配符，则该 SA 既可用于隧道模式，亦可用于传送模式。
- SA 生存期。它规定了每个 SA 最长能够存在的时间，外加一个当该 SA 过期时是被替代还是终止的标识。超出这个时间，该 SA 便不可继续使用，或者被新的 SA 替代，或者被终止。生存期参数有两种形式：可表达成受该 SA 保护的字节数量，也可

表达成 SA 的持续时间，也可以同时用这两种方式来表达一个生存期，以先过期为准。为避免在 SA 过期后造成通信的停顿，可采用两种类型的 SA 生存期软限制和硬限制。所谓“软限制”，是指用它来警告内核，通知它 SA 很快就要到期了，通信的对等实体必须重新协商一个新的 SA 来代替已有的 SA。这样一来，在“硬限制”到期之前，内核便能及时地协商好一个新的 SA。

- 隧道目的地。对于隧道模式中的 IPSec 来说，需用该字段指出隧道的目的地，即外部 IP 报头中的目标地址。
- 路径最大传输单元参数。路径最大传输单元参数（path maximum transfer unit，PMTU）是 IP 报文经过一个特定的、从源主机到目标主机的网络而无须分段的 IP 报文的最大长度。在隧道模式下使用 IPSec 时，必须维持正确的 PMTU 信息，以便对这个数据包进行相应的分段。

图 5.6 给出了一个外出的 IP 报文查询安全策略和安全协定的过程。首先在主机上手工指定 IPSec 通信的安全策略，例如对某个目标地址的 IP 报文使用 ESP 进行加密，但不使用 AH 鉴别协议。该安全策略对应的安全协定（如密钥及其参数）可以手工指定，也可以由 IKE 协商产生。如图 5.6 所示，当一个 SA 协商完成后，这些参数被存放在 SPD 和 SAD 中。

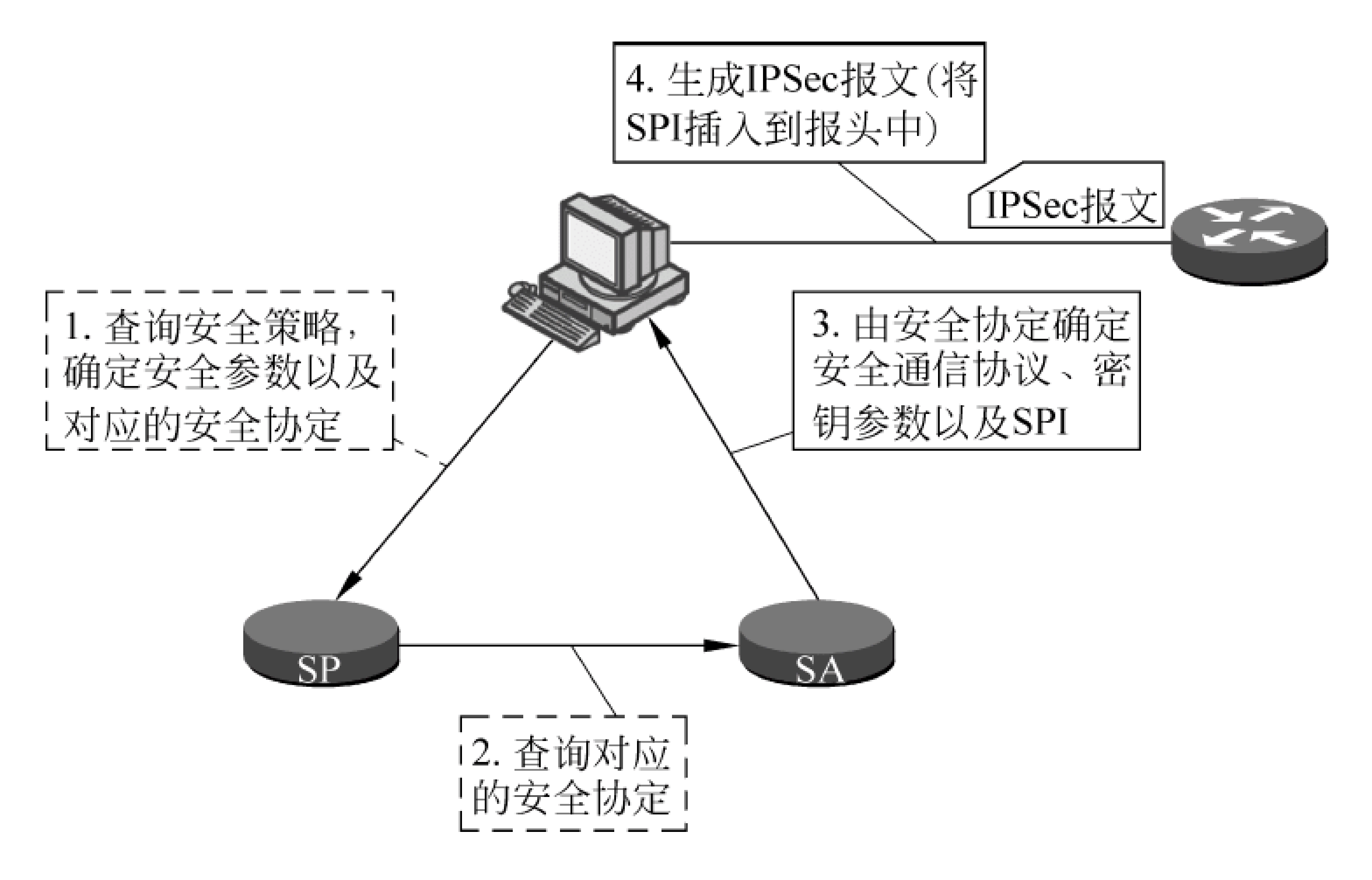


图 5.6 IPSec 输出数据包处理

数据包外出时，IPSec 协议栈根据数据包头部的目标地址、端口确定该数据包对应的 SP，并通过与其关联的 SA 指针找到对应的 SA，然后将 SPI 插入到 ESP 或 AH 的头部字段中。最后，按照相应的工作模式（隧道模式或传输模式）进行 ESP 安装，并将数据包发往目标地址。

对于进入的数据包，IPSec 协议栈根据＜SPI，dst/src，protocol＞三元组确定该 ESP 对应的 SA，从中获取各种参数后，进行相应的验证和解密工作。

5.5　密钥交换协议

Internet 密钥交换协议 IKE 是一个以受保护方式为 SA 协商并提供经过认证的密钥管理协议。使用 IPSec 保护 IP 报文之前,必须先建立一个安全协定 SA。如前所述,SA 可以手工创建或动态建立,IKE 协议用于动态建立 SA。IKE 代表 IPSec 对 SA 进行协商,并对安全协定数据库 SAD 进行填充。IKE 实际上是一种混合型协议。它建立在由 Internet 安全协定和密钥管理协议 ISAKMP 定义的一个框架上。同时,IKE 还实现了两种密钥管理协议的一部分:Oakley 和 SKEME。IKE 是建立在 ISAKMP 基础上的,但两者是不同的:ISAKMP 提供了一个可以由任意密钥交换协议使用的通用密钥交换框架,而 IKE 则定义了一个实际可用的具体的密钥交换协议。

5.5.1　ISAKMP

Internet 安全协定密钥管理协议 ISAKMP 定义协商、建立、修改和删除 SA 的过程和对应的消息格式。ISAKMP 被设计为与密钥交换协议无关的协议,即不受限于任何具体的密钥交换协议、密码算法、密钥生成技术或认证机制。同时,ISAKMP 是作为一个通用的协商协议定义的,而不是仅对 IKE、IPSec 或 IP。

通信双方通过 ISAKMP 向对方提供自己支持的安全功能从而协商共同的安全属性。ISAKMP 消息可以通过 TCP 和 UDP 传输,默认端口为 500。ISAKMP 包括两阶段协商。

- 阶段 1:ISAKMP 通信双方建立一个 ISAKMP SA,它用于保护双方后面的安全协定的协商过程。
- 阶段 2:使用 ISAKMP SA 为 IPSec 安全通信子协议(AH 和 ESP)建立安全协定。

ISAKMP 提供了详细的协议描述和消息格式。ISAKMP 消息使用一系列不同的载荷建立,这些载荷可能在不同的组合中出现。每种载荷控制了一个在密钥协商中的特定数据类型,并且包含一个指向数据包中下一个载荷的指针。通过一系列这种载荷的排列,ISAKMP 包能够包含一个特定 IKE 消息的所有数据。

1. ISAKMP 消息格式

如图 5.7 所示,ISAKMP 消息由一个定长的报头和不定长的载荷组成。定长的报头简化了协议分析过程,它包括协议所需的各种信息来维持状态、处理载荷,并用来提供不可否认性服务和防御重放攻击。

发起者 cookie				
响应者 cookie				
下一个载荷	主版本	次版本	交换类型	标志
消息 ID				
长度				

图 5.7　ISAKMP 报头格式

- 发起者 cookie(initiator cookie)：32 位，可以帮助通信双方确定信息是否来自对方。一个 cookie 对应于一个 SA 建立请求、SA 通告或 SA 删除。
- 响应者 cookie(responder cookie)：32 位，与发起者 cookie 类似，响应者 cookie 用于应答一个 SA 建立请求、SA 通告或 SA 删除。
- 下一个载荷(next payload)：这个 8 位域说明消息中的第一个载荷。
- 主版本(main version)：这个 4 位域指定所用 ISAKMP 协议的版本。
- 次版本(min version)：这个 4 位域包含协议的次版本号。
- 交换类型(exchange type)：这个 8 位域指定组成消息的交换类型。
- 标志(flag)：这个 8 位域说明为 ISAKMP 设置的具体选项。目前使用了这个域的前三位，分别是加密位、提交位和认证位。其他位在传输前被设置为 0。
- 消息 ID：4 个字节，用来标识一个 ISAKMP 消息。
- 长度：4 个字节，以 8 位字节来计算的整个消息(报头加载荷)的长度。

2. ISAKMP 载荷

在 ISAKMP 定长的报头后面是不定长的载荷。RFC 2408 共定义了 13 种载荷。表 5.1 说明了这些载荷所分配的值，这些值会出现在 ISAKMP 头的下一个载荷字段中或者是载荷头中下一个载荷字段中。

表 5.1　ISAKMP 载荷分配表

下一个载荷	分配的值	下一个载荷	分配的值
无	0	杂凑载荷	8
安全协定载荷	1	签名载荷	9
建议载荷	2	Nonce 载荷	10
交换载荷	3	通知载荷	11
密钥交换载荷	4	删除载荷	12
标识载荷	5	厂商载荷	13
证书载荷	6	保留	14～127
证书请求载荷	7	私有使用	128～255

5.5.2 IKE

IKE 的密钥交换分为两个独立阶段：第一个阶段建立一个安全通道，使得第二个阶段的协商可以秘密地进行；第二个阶段为 IPSec 创建安全协定。

在第一阶段，通信双方彼此建立一个已通过身份认证和安全保护的隧道，称为 ISAKMPSA（IKE 的一种安全协定，也称为 IKESA）。一旦 ISAKMPSA 建立起来，所有发起方与应答方之间的 IKE 通信都经过加密、完整性检查和认证保护。

两台主机之间可以同时建立多个 ISAKMPSA，一个 ISAKMPSA 也可以用于创建多个 IPSec SA，ISAKMPSA 的结束不会影响其创建的 IPSec SA 发生作用。

IKE 定义了几组用于 Diffie-Hellman 交换的群参数，并提供用户创建新群的机制。

IKE 定义了主模式、野蛮模式、快速模式和新群模式 4 种交换模式。前三个用于协商 SA，第四个用于协商 Diffie-Hellman 交换的群。SA 或群提议（group offer）以变换载荷的形式封装在建议载荷中，而建议载荷又封装在安全载荷中。下面分别介绍这些交换模式。

1. 主模式

主模式用于协商 IKE 密钥交换阶段 1 的 ISAKMPSA，它包括一个经过认证的 Diffie-Hellman 密钥交换过程。主模式将密钥交换信息与用户身份、认证信息分离，这种分离保护了身份信息，因为交换的身份信息受到了前面生成的 Diffie-Hellman 共享秘密的保护。图 5.8 说明了主模式下的消息交互过程。

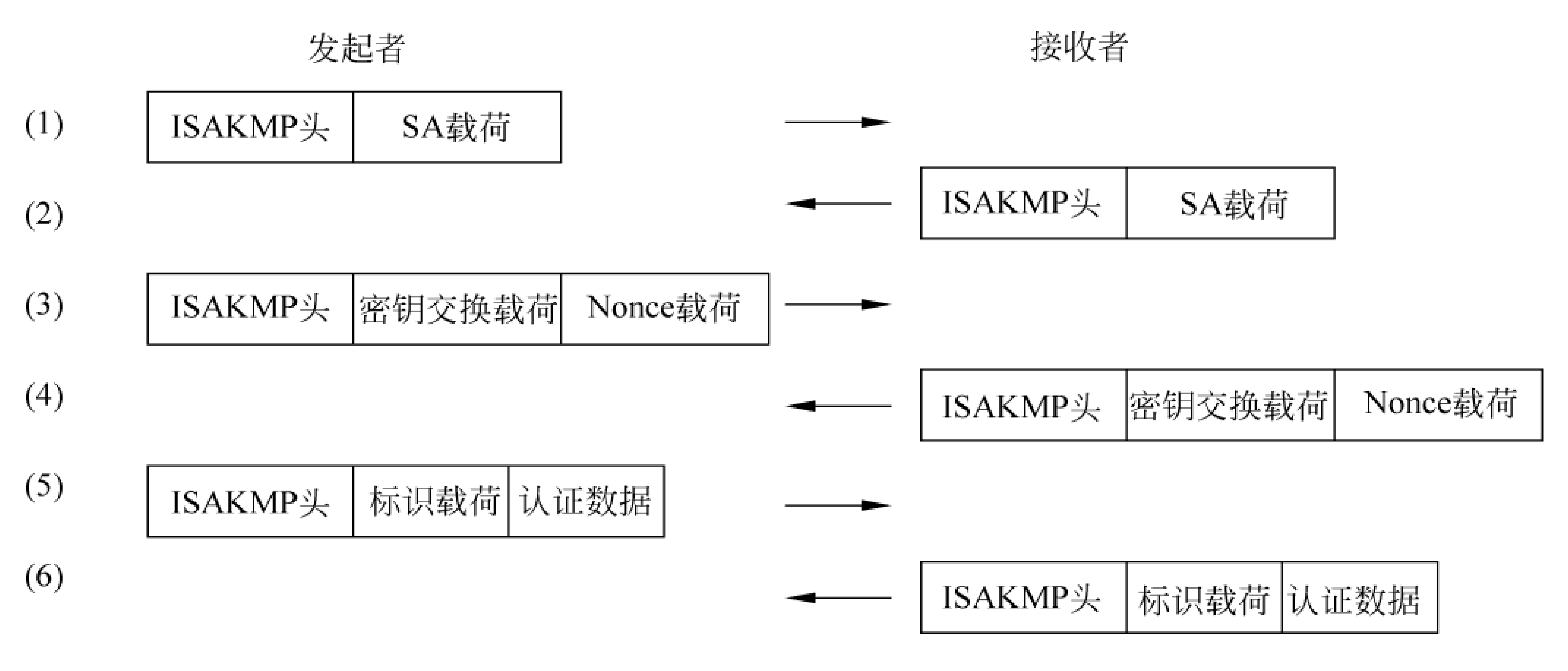


图 5.8 IKE 主模式交换

- 消息 1：发起者向响应者发送一个封装有建议载荷的 SA 载荷，而建议载荷中又封装有变换载荷。
- 消息 2：响应者发送一个 SA 载荷，该载荷表明它所能够接受的正在协商的 SA 的

建议。

- 消息 3 和消息 4：发起者和响应者交换 Diffie-Hellman 公开值和辅助数据。这是计算共享秘密(用来生成加密密钥和认证密钥)所必需的。
- 消息 5 和消息 6：发起者和响应者交换标识数据并认证 Diffie-Hellman 交换。这两个消息中传递的信息是加密的，用于加密的密钥使用消息 3 和消息 4 中交换的密钥信息生成，因此用户身份信息受到了保护。

2. 野蛮模式

在不需要保护身份信息时，IKE 使用野蛮模式来协商阶段 1 的 SA。野蛮模式允许同时传送与 SA、密钥交换和认证相关的载荷。将这些载荷组合到一条消息中减少了消息的往返次数，但是这样无法提供身份保护。图 5.9 显示了野蛮模式下的消息交互过程。

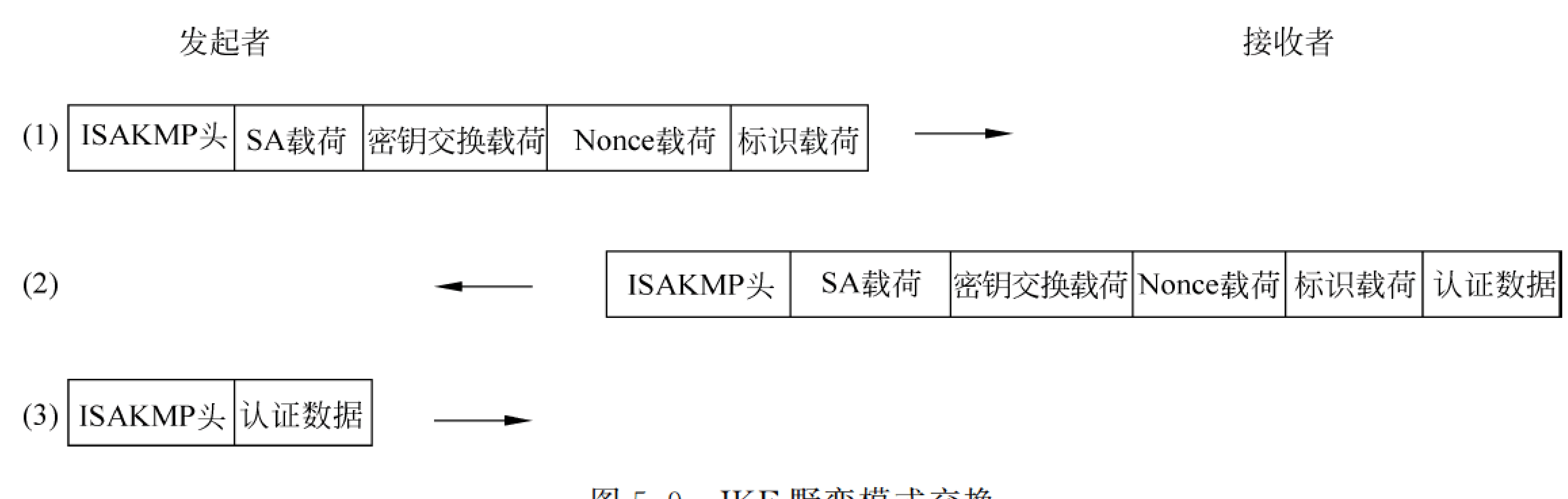


图 5.9 IKE 野蛮模式交换

- 消息 1：与主模式类似，发起者向响应者发送一个封装有单个建议载荷的 SA 载荷，而建议载荷中又封装有一个变换载荷。但在野蛮模式中，只提供带有一个变换的建议载荷；响应者可以选择接收或拒绝该建议。Diffie-Hellman 公开值、需要的随机数和身份信息也在第一条消息中传送。
- 消息 2：如果响应者接收发起者的建议，它发送一个 SA 载荷，其中封装有发起者建议的变换的建议载荷。它将 Diffie-Hellman 公开值、需要的随机数和身份信息作为消息的一部分同时传送。这个消息受到协商一致的认证函数保护。
- 消息 3：发起者发送经过双方一致同意的哈希函数生成的散列值。这个消息可以认证发起者的身份并且证明其为交换的参与者。这个消息使用前两个消息交换的密钥信息生成的密钥进行加密。需要注意的是，包含身份信息的消息未被加密，所以和主模式不同，野蛮模式不提供身份保护。

3. 快速模式

快速模式用于协商阶段 2 的 SA，协商受到在阶段 1 协商好的 ISAKMPSA 的保护。在

快速模式下交换的载荷都是加密的。快速模式下的消息交互过程如图 5.10 所示。

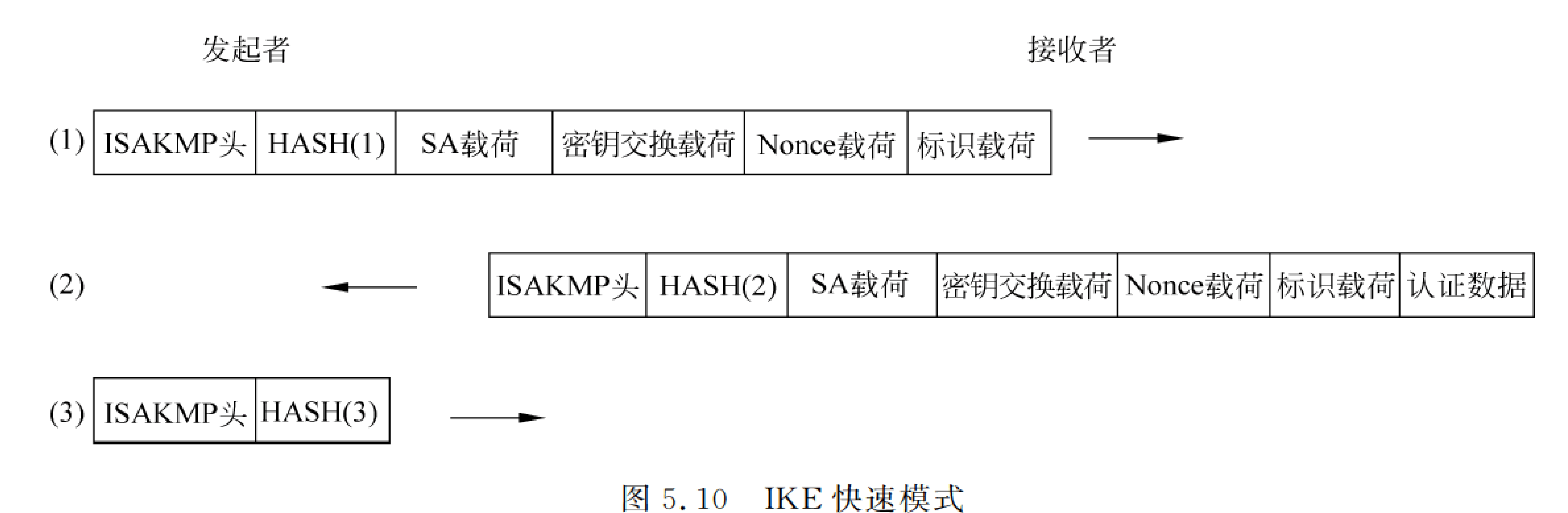


图 5.10 IKE 快速模式

- 消息 1：发起者向接收者发送一个杂凑载荷、一个 SA 载荷(其中封装了一个或多个建议载荷，而每个建议载荷中又封装一个或多个变换载荷)、一个 Nonce 载荷，可选的密钥交换信息和标识信息。杂凑载荷中包含消息摘要 HASH(1)，它是使用前面协商好的伪随机函数对消息头中的消息 ID(MSgID)连同杂凑载荷(哈希值)的全部消息部分(包括所有的载荷头)进行计算的结果。
- 消息 2：这个消息中的载荷和消息 1 中的载荷类似。HASH(2)中包含的散列值的生成方法和 HASH(1)中类似，只是除去了载荷头的发起者 Nonce。
- 消息 3：这个消息对于前面的交换进行认证，它仅由 ISAKMP 头和杂凑载荷组成。杂凑载荷中的消息摘要 HASH(3)是以一个为 0 的字节连接着 MSgID，以及去掉了载荷头的发起者 Nonce 和载荷头的响应者 Nonce 而生成的。

4. 新群模式

新群模式用于为 Diffie-Hellman 密钥交换协商一个新的群。新群模式是在 ISAKMP 阶段 1 中交换的 SA 的保护之下进行的。图 5.11 是这种模式下的消息交换过程。



图 5.11 IKE 新群模式

在第一条消息中，发起方发送一个 SA 载荷，其中包含了新群的特征(例如模指数运算的质数)，如果响应者能够接受这个群，便在第二条消息中使用完全一样的信息做出应答，否

则拒绝该提议。

5.5.3 IKE 在 IPSec 中的应用

下面举例说明两台主机之间使用 IKE 协商 SA 以及进行 IPSec 安全通信的过程。

如图 5.12 所示，假设两台主机 A 和 B 均定义了 IPSec 安全策略，并启动 IPSec 进行安全通信，IPSec 工作在传输模式。主机上的 IPSec 安全策略通过 IP 筛选器列表定义，该列表中定义了源地址、目标地址、要保护的上层协议和端口号等，通过设置这些参数确定某个安全策略，然后对该安全策略设置对应的安全协定。数据包外出时，通过将数据包的这些参数和筛选器列表相匹配来确定该数据流对应的安全策略和安全协定。

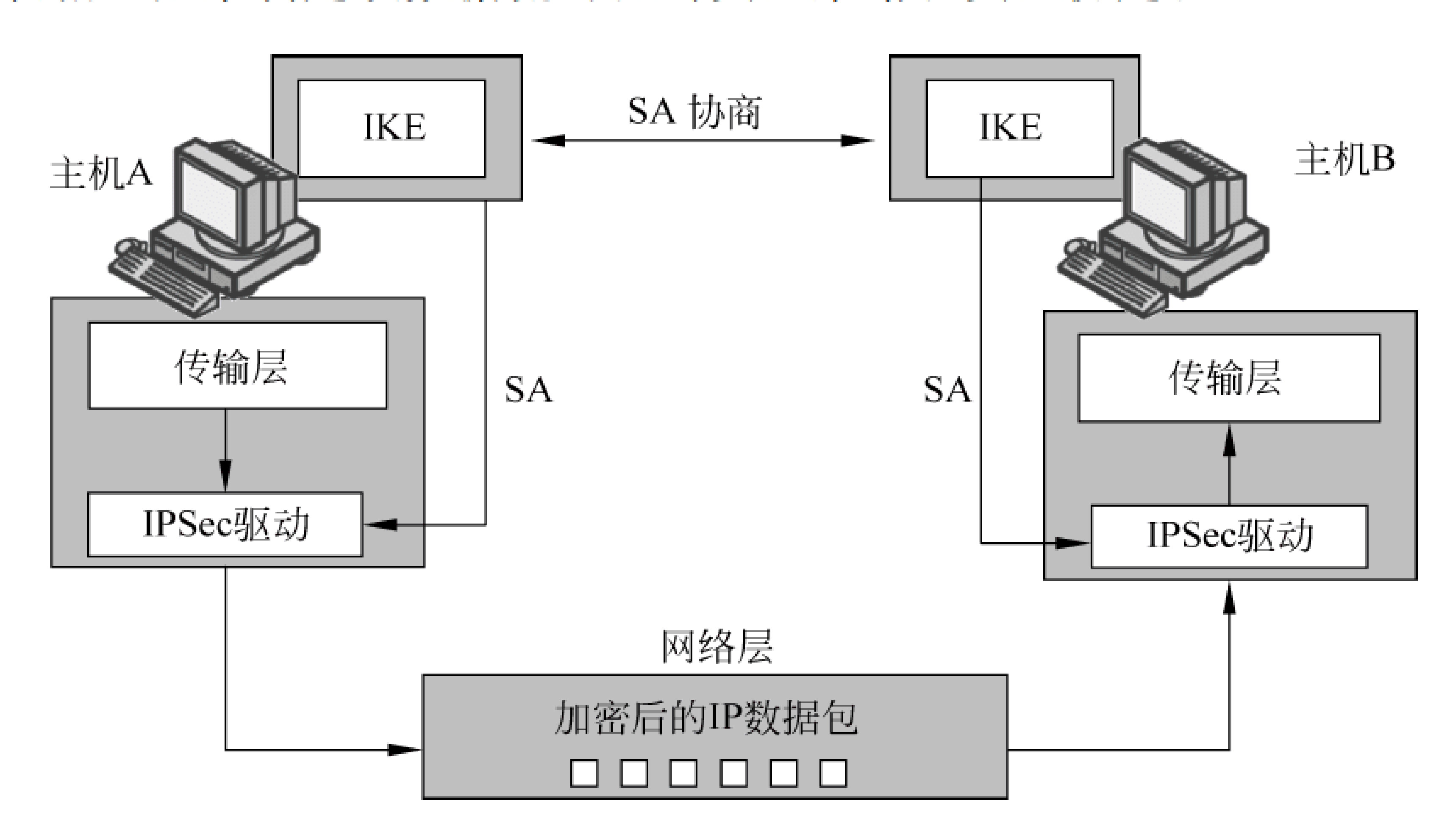


图 5.12 IKE 在 IPSec 中的应用

这种情形下，两台主机使用 IPSec 进行安全通信的过程可描述如下。

① 主机 A 上的应用程序向主机 B 发送一个包含应用层协议的 IP 报文。

② 主机 A 上的 IPSec 驱动程序检查其出站 IP 筛选器列表，确定应该保护该数据包。

③ 为了协商安全协定，IPSec 驱动程序通知 IKE 开始进行密钥协商。主机 A 上的 IKE 服务将自己的 IP 地址用作源地址，主机 B 的 IP 地址用作目标地址完成策略查找，主模式筛选器匹配确定了主机 A 向主机 B 提供的主模式设置；主机 A 在主模式下，使用 UDP 源端口 500、目标端口 500 发送第一条 IKE 消息。IKE 报文通过 IPSec 驱动程序接受特殊处理，以跳过筛选器。

④ 主机 B 收到请求安全协商的 IKE 主模式消息(参见 5.5.2 节)，它使用 UDP 报文的源 IP 地址和目标 IP 地址执行主模式策略查找，以确定要同意哪些安全设置。主机 B 具有匹配的主模式文件，因此进行回复，以开始主模式 SA 的协商。

⑤ 主机 A 与主机 B 开始协商选项、交换标识、验证这些标识的信任度(身份验证)，然

后生成共享的主密钥。它们现在已经建立了 IKE 主模式 SA。主机 A 与主机 B 相互信任。

⑥ 主机 A 执行 IKE 快速模式策略查找,选择快速模式安全设置并向主机 B 提供这些设置以及快速模式筛选器。

⑦ 主机 B 使用主机 A 提供的筛选器描述执行 IKE 快速模式策略查找。主机 B 选择其策略所需的安全设置,并将这些设置与主机 A 提供的设置进行比较。主机 B 将接受一组选项,并完成其余的 IKE 快速模式协商步骤,以创建一对 IPSec 安全协定。

此后,该 IPSec SA 为应用程序的数据通信提供透明的保护。只要应用程序发送和接收数据,IPSec SA 就会自动被 IKE 快速模式协商刷新。当应用程序停止发送和接收数据时,IPSec SA 即处于空闲状态,并被删除。

通常,IKE 主模式 SA 不会被删除。默认情况下,主模式 SA 的生存期为 8 小时。当发送数据的通信量比较大时,IKE 将自动协商新的快速模式,以创建两个新的 IPSec SA 来保护应用层协议的通信。这种协商的主模式 SA 已经存在,因此协商过程不会耗费太多时间。如果主模式 SA 过期,则 IKE 会根据需要自动重新协商主模式 SA。

通信的主机之间的路径中,所有路由器或交换机将加密的 IP 报文转发给它们的目的地。但是,如果路径中有防火墙、安全路由器或代理服务器,就可能不会转发 IPSec 与 IKE 通信。必须配置这些设备以允许 IPSec 和 IKE 协议数据包经过。如果 IPSec 报文未加密,防火墙或安全路由器仍可以检查 TCP 或 UDP 端口或数据包中的其他内容。如果这些数据包的内容在传输过程中被修改,那么接收 IPSec 的主机就会检测出这种修改并丢弃这些数据包。

5.6 验证头 AH

如前所述,AH 协议可以提供无连接的数据完整性、数据源认证和抗重放保护等安全服务。AH 使用消息验证码 HMAC 对 IP 报文进行认证。AH 协议定义保护方法、AH 报头的位置、身份鉴别的覆盖范围以及输入和输出处理规则,但不对所用身份验证算法和验证方法进行定义,这取决于 IPSec 的协议实现。例如 Windows 系统中,IPSec 可提供三种形式的报文源鉴别:基于对称密钥(共享密钥)的、基于非对称密钥(使用数字证书)的以及 Kerberos 认证。AH 没有硬性规定必须使用抗重放保护,是否使用该服务由接收端自行处理。

5.6.1 AH 报文格式

如图 5.13 所示,AH 报头由 5 个固定长度域和一个变长的认证数据域组成。下面分别描述各个域的功能。

发起者 cookie				
响应者 cookie				
下一个载荷	主版本	次版本	交换类型	标志
消息 ID				
长度				

图 5.13 认证头格式

- 下一个头(next header):这个 8 比特的域指出 AH 后的下一个载荷的类型。在传输模式下,“下一个头”是被保护的上层协议,如 UDP 或 TCP 协议。在隧道模式下,AH 的上层协议则是原始 IP 报文,例如其值为 4 表示 IP—IP(IPv4)封装。如果后面是另一个 AH 载荷(用于嵌套使用的 IPSec),则这个域值为 51。该域的取值如表 5.2 所示。

表 5.2 AH 报文中的 Next Header 的取值

取 值	关 键 字	协 议
0		Reserved
1	ICMP	Internet Control Message
2	IGMP	Internet Group Management
4	IP	IP in IP(encasulation)
6	TCP	Transmission Control
17	UDP	User Datagram
50	SIPP-ESP	SIPP Encap Security Payload
51	SIPP-AH	SIPP Authentication Header

- 载荷长度(payload length):这个 8 比特的域包含以 2 比特为单位的 AH 的长度减 2。因为 AH 头是一个 IPv6 扩展头,按照 RFC2460,它的长度是从 64 比特表示的头长度中减去一个 64 比特得到的。但 AH 采用 32 比特来计算,因此,计算时应该减去两个 32 比特(或一个 64 比特)。
- 保留(reserved):这个 16 比特的保留域供将来使用。AH 规定这个域应被置为 0。
- 安全参数索引(SPI):SPI 是一个 32 比特的证书,用于和 IP 报文的源地址或目的地址以及 IPSec 协议(AH 或 ESP)共同唯一标识一个数据包所属的数据流的安全协定 SA。SPI 域的取值中,1～255 被留作将来使用,0 被保留。
- 序列号(sequence number):这个域包含一个作为单调增加的计数器的 32 位无符号整数。AH 使用序列号和滑动窗口来提供抗重放安全服务。SA 建立时,发送方和接收方序列号初始化为 0;通信双方每使用一个特定的 SA 发送一个数据包则将其值加 1,用于抵抗重放攻击;AH 规范强制发送者必须发送序列号给接收者,而接收者可以选择不使用抗重放特性,这时它不理会该序列号即可;若接收者启用抗重放特性,则使用滑动窗口机制检测重放包。具体采用的滑动窗口协议因 IPSec 的实现而异。

- 认证数据(authentication data):这个变长域包含数据包的认证数据,该认证数据被称为数据包的完整性校验值(integrity check value,ICV)。用来生成 ICV 的算法由 SA 指定,用来计算 ICV 的可用的算法因 IPSec 的实现不同而不同。为了保证互操作性,AH 强制所有的 IPSec 实现必须包含两个 MAC,即 HMAC-MD5 和 HMAC-SHA-1。ICV 的长度依赖于所使用的 MAC 算法,例如 HMAC-MD5 可以为 128 位,HMAC-SHA-1 可以为 160 位。对于 IPv4 报文,这个域的长度必须是 32 的整数倍;对于 IPv6 报文,这个域的长度必须是 64 的整数倍。如果一个 IPv4 报文的 ICV 域的长度不是 32 的整数倍,或者一个 IPv6 报文的 ICV 域的长度不是 64 的整数倍,必须添加填充位使 ICV 域的长度达到所需长度。

5.6.2 AH 操作模式

在进行安全通信保护时,AH 头的位置依赖于 AH 的操作模式。AH 有两种操作模式,即传输模式和隧道模式。

1. AH 传输模式

在传输模式中,AH 报头被插入在 IP 报文中,紧跟在 IP 头之后和需要保护的上层协议或其他 IPSec 协议头之前(用于 IPSec 的嵌套)。因此,在传输模式中,对 IPv4 而言,AH 被插在 IP 变长可选域之后。图 5.14 说明了 IPv4 的传输模式下 AH 相对于其他头部域的位置。



图 5.14　IPv4 传输模式下 AH 的封装

IPv6 中不再有以往的 IP 报头中的可选域。IPv6 中,选项被处理为单独的头,称作扩展头。扩展头插入在 IP 报头的后面,这个特征提高了 IP 报文的处理速度:除了包含路由器需要检查的路由信息的逐跳(hop by hop)扩展头外,其他头部字段不会被从源到目的的沿途中间节点处理,而是仅被目标主机处理。

在 IPv6 的传输模式中,AH 被插入在逐跳、路由和分段扩展头的后面;目的选项扩展头可以被置于 AH 头的前面或后面。如果目的选项头被出现在 IPv6 目的地址域的第一个

目标主机以及其后的路由头中列出的目标主机列表处理，那么它仅被目标主机处理，则应该放在 AH 之后。图 5.15 说明了 IPv6 中，AH 传输模式下，AH 报头相对于其他 IPv6 扩展头的位置。

应用AH之前的IPv6报文

IP报头	扩展头(如果有)	传输协议头	传输协议数据

应用AH之后的IPv6报文

IP报头	逐跳、路由分段以及目的扩展头	AH认证头	传输协议头	传输协议数据

←认证范围→

图 5.15 IPv6 传输模式下 AH 的封装

无论在 IPv4 下还是在 IPv6 下，AH 验证的都是整个 IP 报文(包括报头)。

2. AH 隧道模式

如图 5.16 所示，在隧道模式中，AH 将自己保护的数据包(通常是 IPv4 或 IPv6 报文)封装起来，然后在 AH 头之前添一个新的 IP 报头。“里面的”IP 报文中包含了通信的原始地址，“外面的”IP 报文则包含了 IPSec 端点的地址。

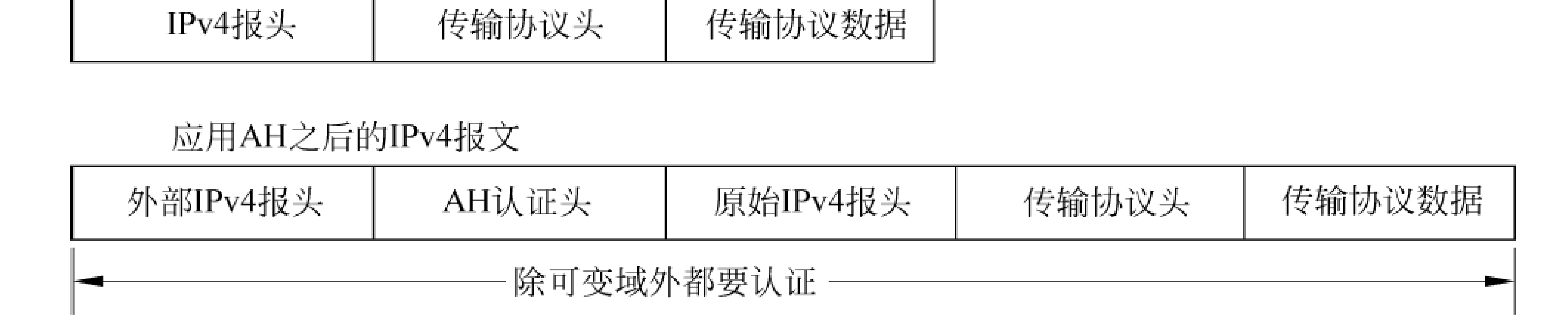


图 5.16 IPv4 隧道模式下 AH 的封装

如图 5.17 所示，在 IPv6 报文中，除了新的 IP 报头外，原始数据包的扩展头也被插入在 AH 报头前面。

应用AH之前的IPv6报文

IP报头	扩展头(如果有)	传输协议头	传输协议数据

应用AH之后的IPv6报文

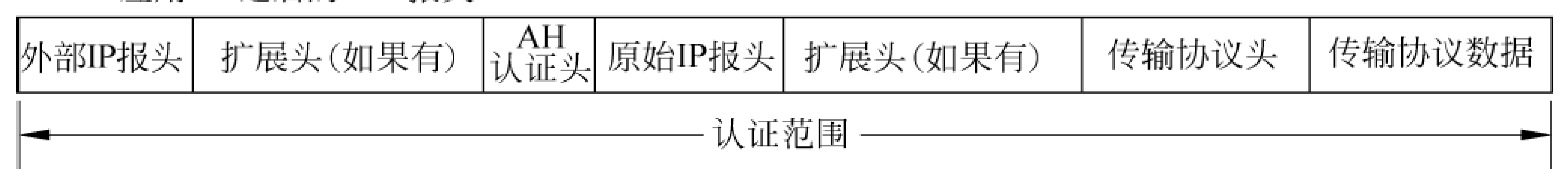


图 5.17 IPv6 隧道模式下 AH 的封装

和传输模式类似，隧道模式下 AH 验证整个 IP 数据包，包括新的 IP 报头。AH 隧道模式可用来替换端到端安全服务的传输模式，但由于 AH 协议没有提供机密性保护，因此 AH 只能保证收到的数据包在传输过程中没有被篡改，并且数据包来源于一个可信的实体，同时它不是一个被重放的数据包。

5.6.3 AH 协议处理过程

图 5.18 中给出了 AH 协议的处理过程。对于外出的数据包(图 5.18 的左侧部分)，AH 协议根据数据包的目标 IP 地址、目标端口号以及协议类型(AH 或 ESP)在 SPD 中查找对应的安全策略，如果没有该数据包对应的策略则将该数据包丢弃，否则，按照 SP 中对应的 SA 指针查找 SAD，确定该数据包对应的 SA。

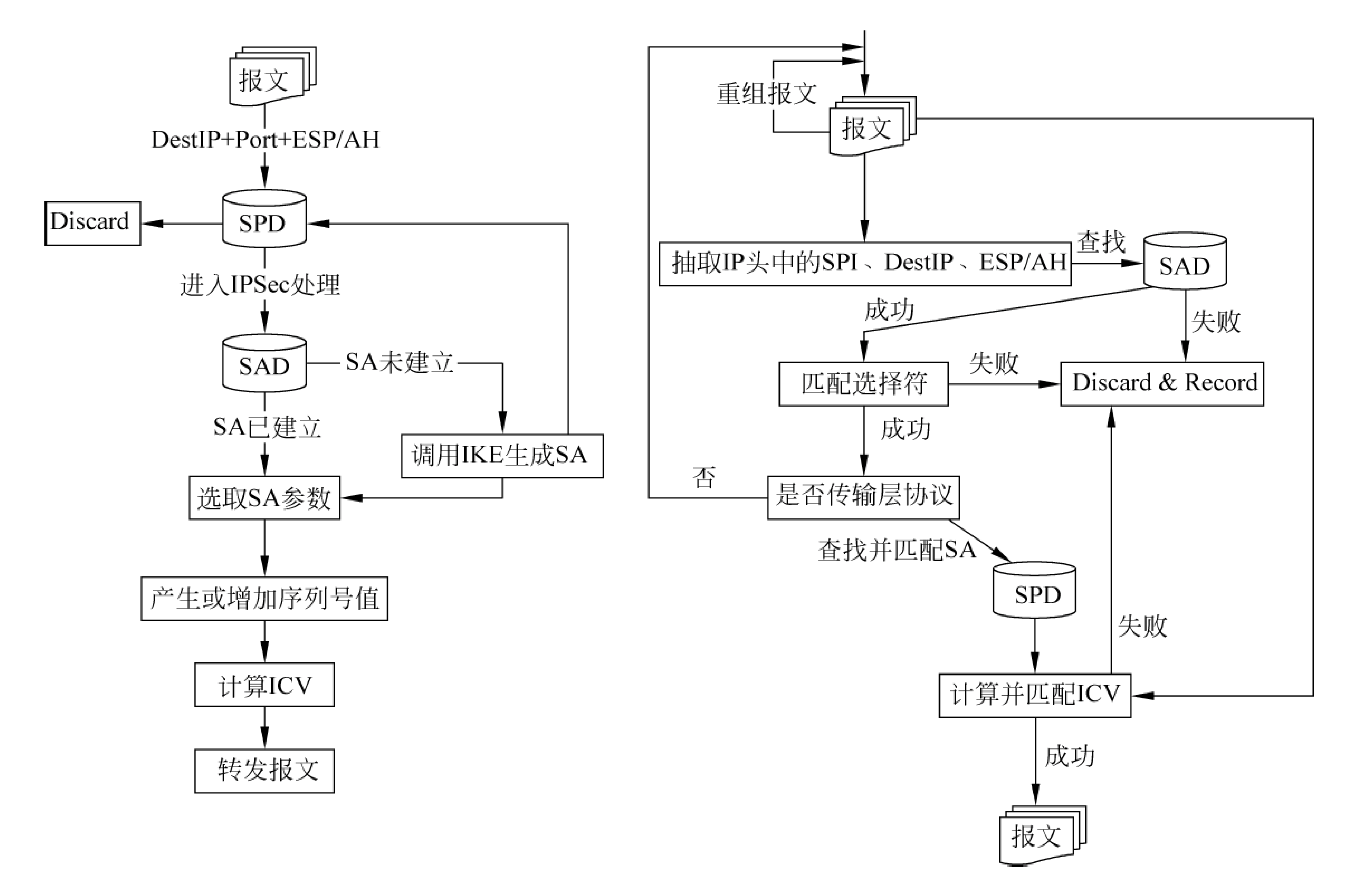


图 5.18 AH 输出-输入处理流程

如果 SA 已建立，则选取 SA 参数后，进入 AH 协议处理过程，对该数据包实施 AH 封装和处理：首先，为了提供抗重放功能，将 AH 报头的序列号值加 1(对于第一个数据包则将其初始化为 0)，其后，根据 SA 中指定的完整性算法和密钥计算 ICV 作为 AH 报文中的验证数据字段，同时将 SPI 的值插入到安全协议的头部；最后，根据 SA 中确定的协议操作

模式，将 AH 报头的其他字段按照操作模式对应的顺序进行协议封装后，将数据包交由链路层协议处理。

如果该数据包对应的 SA 未建立，则调用 IKE 生成 AH 所需的安全参数（如密钥等），然后选取对应的 SA 后，进入 AH 协议处理过程。

对于进入的数据包（图 5.18 的右侧部分），考虑到 IP 报文可能会被分片，IPSec 首先按照 IP 协议的规定对数据包进行重组，还原为经过鉴别保护的 IP 报文后，进行 AH 协议处理：从 AH 数据包的头部提取 SPI、目标 IP 地址以及协议（AH）等字段，将该三元组作为索引值查找 SAD 数据库，以确定该数据包对应的安全参数；如果成功找到该数据包对应的 SA，并进行 IP 报文重组后，按照 SA 中的参数计算 ICV，将之和 AH 数据包中的认证数据比较，匹配成功后将该报文还原为 IP 报文，交由上层协议处理；如果没有对应的 SA，或 ICV 校验不成功，则将该数据包丢弃。

5.7 封装安全载荷 ESP

ESP 可以提供和 AH 类似的服务，但增加了两个额外的服务：数据机密性和有限的数据流保密服务。其中，数据机密性服务通过使用对称密码加密 IP 报文的相关部分实现，数据流保密服务由隧道模式下的机密性服务提供。ESP 加密数据包的密码算法取决于 IPSec 实现以及用户的安全协定。例如，加密算法可以是 3DES、DES 等。和 AH 类似，ESP 使用消息验证码 MAC 提供认证服务，并且产生消息鉴别码时需要一个密钥，即使用 HMAC。ESP 提供的机密性和数据源鉴别（包括完整性保护）功能是可选的，例如用户可以选择只加密不鉴别，也可以同时使用加密和鉴别服务。ESP 既可以单独使用，也可以以嵌套的方式使用或者和 AH 结合使用。

5.7.1 ESP 报文格式

如图 5.19 所示，ESP 报文由 4 个固定长度的域和三个变长域组成。以下分别描述各个域的功能。

- 安全参数索引（security parameters index，SPI）：同 AH 的 SPI 一样，ESP 的 SPI 也是一个 32 比特的索引值，用于和源地址或目的地址以及 IPSec 安全通信协议（AH 或 ESP）共同唯一标识一个数据包所属的安全协定。SPI 本身需要经过验

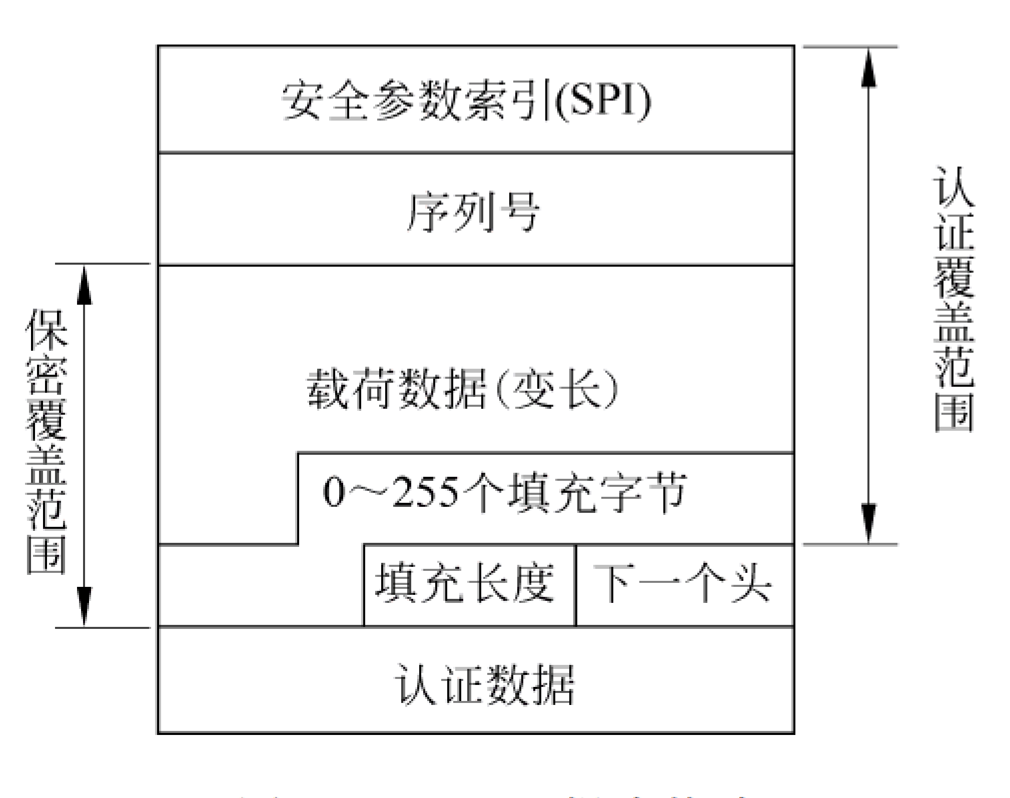


图 5.19 ESP 报头格式

证，但不进行加密。如果 SPI 本身被加密，则无法得到解密 ESP 数据包的各种参数（如解密算法和密钥等），因为它们保存在 SA 中。

- 序列号（sequence number）：同 AH 一样，这个域是一个作为单调增加的计数器的 32 位无符号整数。ESP 使用序列号和滑动窗口来提供抗重放的安全服务。同 SPI 一样，序列号需要经过验证（在选择了带验证功能的 ESP 的情况下），但不进行加密。这是由于 ESP 将根据它判断一个包是否重复，对于重复的包无须解密即可简单丢弃，因此序列号没有必要保密，只要可以证明它没有被篡改就可以了。
- 载荷数据（payload data）：这是一个变长域。这个字段包含了 ESP 保护的实际数据。因此，这个字段的长度由 IPSec 净载荷的数据长度决定。它的长度以比特为单位并且必须是 8 的整数倍。也可在保护数据字段中包含一个加密算法可能需要用到的初始化向量 IV，但初始化向量不加密。
- 填充（padding）：该字段用于在 ESP 中保证边界的正确。长度是 0～255 字节。某些加密算法模式要求密码的输入是其块大小的整数倍，填充项就是用来完成这一任务的。同时，假如 SA 没有要求机密性保证，仍需通过填充项把 ESP 头的“填充长度”和“下一个头”这两个字段靠右排列。此外，这项技术还可用来隐藏载荷数据的真正长度。至于具体填充内容，与提供机密性的加密算法有关。
- 填充长度（pad length）：填充长度是一个 8 比特的域，表明填充域中填充比特的长度。接收端可以根据填充长度恢复载荷数据的真实长度。这个域的有效值是 0～255 字节。填充项长度字段是硬性规定的，因此，即使没有填充，填充长度字段仍会将它表示出来。
- 下一个头（next header）：和 AH 类似，这个 8 比特的域表明了载荷中封装的数据包类型。如果是传输模式，“下一个头”可能是一个传输层协议的值（如 TCP 对应的值是 6）；如果是隧道模式，则是原始 IP 报头的协议值（如 IPv4 对应的值是 4）；如果是嵌套方式，则可能是 50 或 51，表示后面是另一个 IPSec 报文（ESP 或 AH）。
- 认证数据（authentication data）：这个变长域中存放 ICV，用于提供数据源鉴别和完整性保护服务。它是对除认证数据域以外的 ESP 报文进行计算获得的。但认证的对象不包括原始 IP 报头，这一点和 AH 不同。ICV 的长度由 SA 所用的认证算法决定。如果对 ESP 报文进行处理的 SA 中没有指定身份验证器，就不会有验证数据字段（即双方协商只使用 ESP 进行加密保护而不进行数据源鉴别）。

5.7.2 ESP 操作模式

和 AH 一样，ESP 在 IP 报文中的位置取决于 ESP 的操作模式，即传输模式和隧道模式。

1. ESP 传输模式

在传输模式下，ESP 被插入到 IP 报头和所有的选项之后、传输层协议之前，或者在已应用的任意 IPSec 协议(如另一个 ESP)之前。在 IPv4 传输模式下，ESP 被插入在 IPv4 的变长选项域之后。图 5.20 显示了 IPv4 中，ESP 在传输模式下相对于其他头部域的位置。

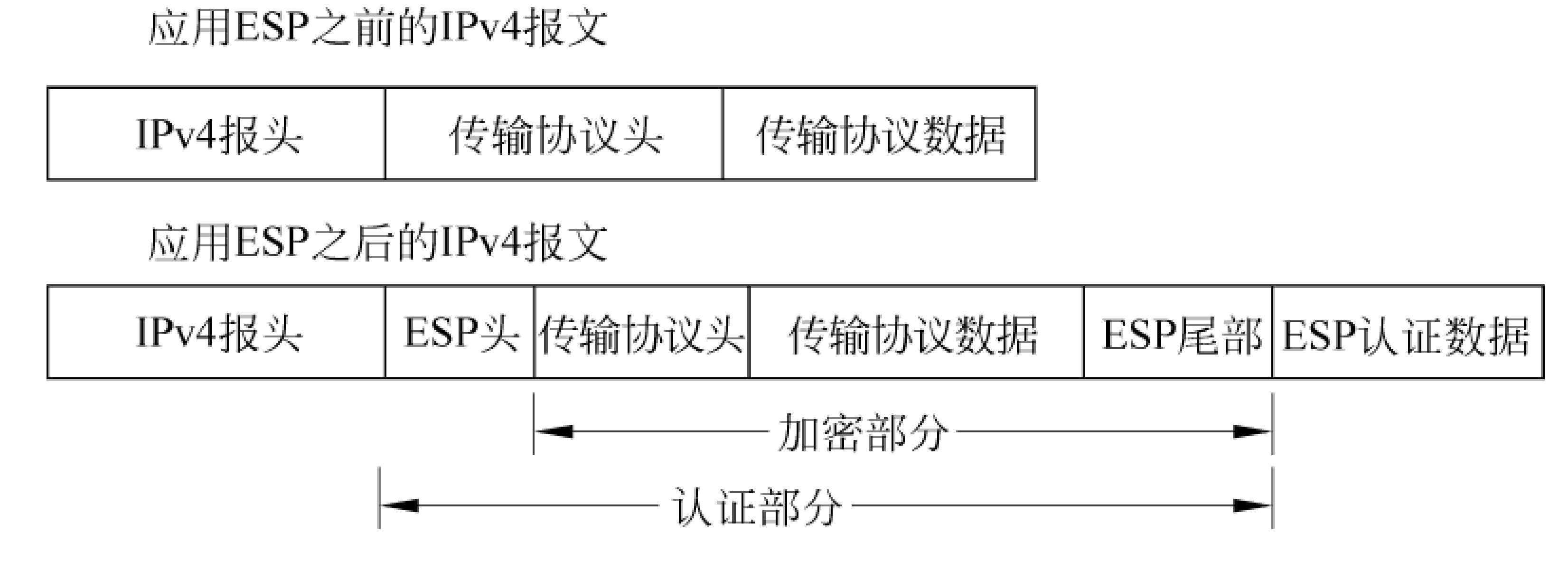


图 5.20　传输模式下 ESP 相对于其他 IPv4 报头的位置

图中的 ESP 头部域由 SPI 和序列号域组成，ESP 尾部由填充域、填充长度域和“下一个头”域组成(如图 5.19 所示)。图中 ESP 同时提供机密性保护和数据认证服务。如前所述，即使需要机密性服务，SPI 和序列号域也不被加密，因为接收节点需要这些域来标志处理数据包的 SA；此外，如果启动了抗重放服务，序列号域还要被用来检验重放数据包。同样，认证数据域也不被加密，因为目标主机在处理这个数据包之前需要使用认证域(ESP 认证数据)来验证数据包的完整性以及消息的来源。

对于 IPv6 报文，ESP 被插入在逐跳、路由和分段扩展头之后；目的选项扩展头可以放在 ESP 报头的前边或后边。如果目标选项头仅被目的节点处理，那么它可以置于 ESP 之后；否则，目标选项头将放在 ESP 之前。图 5.21 说明了 IPv6 传输操作模式下，ESP 相对于其他 IPv6 扩展头的位置。

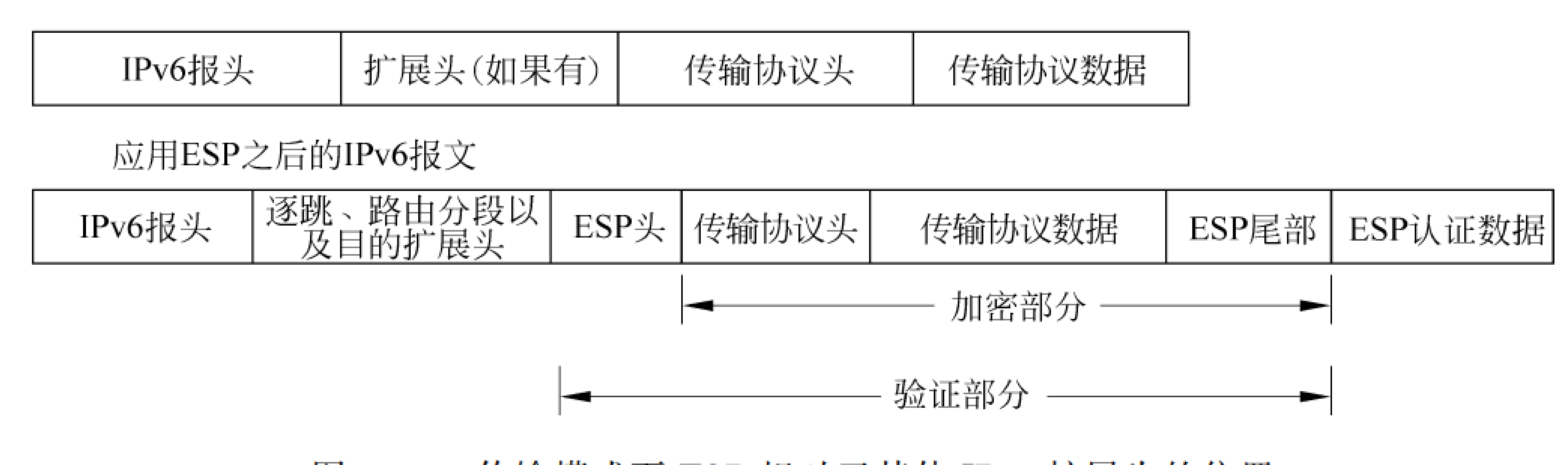


图 5.21　传输模式下 ESP 相对于其他 IPv6 扩展头的位置

从图 5.20 和图 5.21 中可以看出，传输模式下，ESP 不对 IP 报文头部进行验证，该部分的验证可以由 AH 实施。由于 ESP 不对 IP 报头进行验证，所以目标主机可能无法检测到 IP 报头发生的修改。这样，ESP 传输模式认证服务所提供的安全性就不如 AH 传输模式。因此，需要更高安全级并且通信双方使用公开 IP 地址时，应该采用 AH 认证服务或联合使用 AH 和 ESP。此外，ESP 在传输模式下是不能提供数据流保密服务的，因为源和目的 IP 地址均未得到加密。

2. ESP 隧道模式

隧道模式下，ESP 头部被插入在原始 IP 报头之前，并且生成一个新的 IP 报头置于 ESP 报头之前。其中，内部 IP 报头的源地址和目的地址是实际的源和目标主机的地址，而外部 IP 地址是 ESP 端点的地址（例如源节点和目的节点的安全网关的地址）。因此，内、外部的 IP 报头的源地址和目的地址可能不同。图 5.22 和图 5.23 中分别给出了 ESP 隧道模式在 IPv4 和 IPv6 中的报文封装顺序。



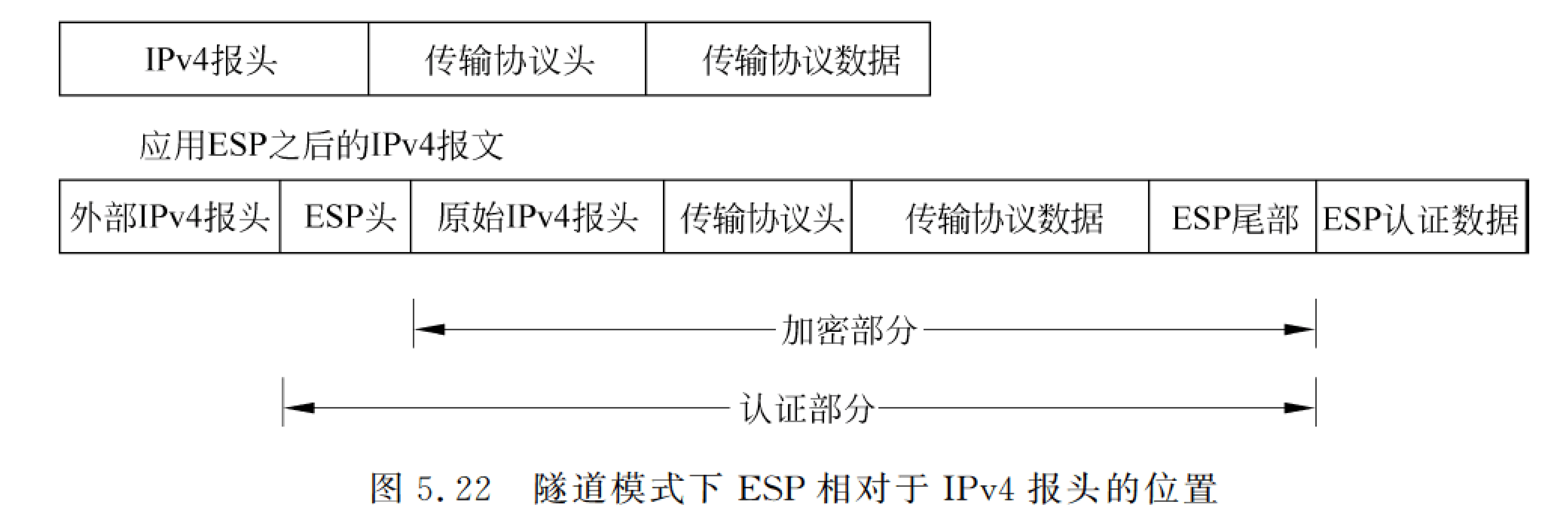


图 5.22 隧道模式下 ESP 相对于 IPv4 报头的位置

应用ESP之前的IPv6报文

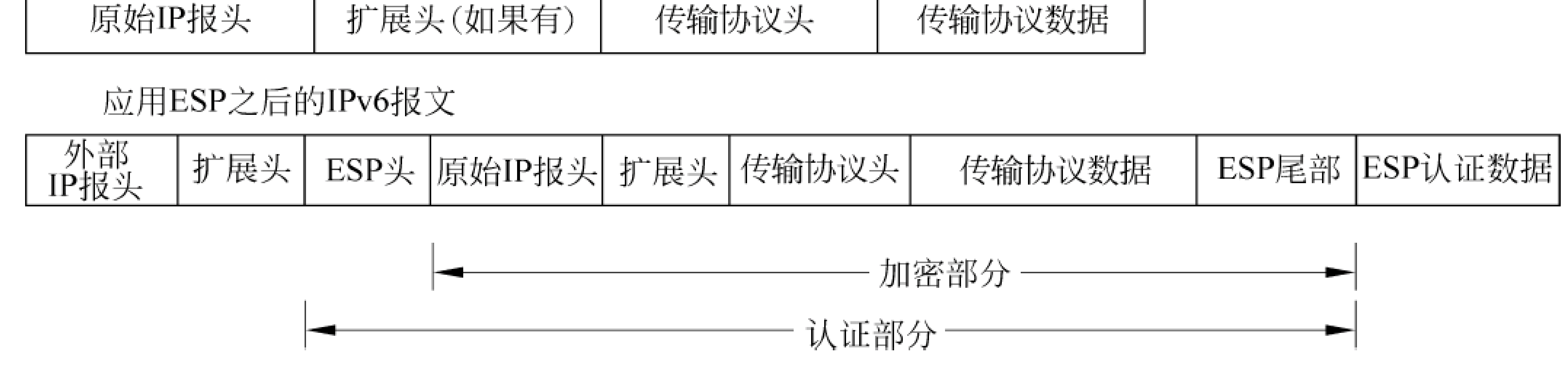


图 5.23 隧道模式下 ESP 相对于 IPv6 及其扩展头的位置

和传输模式类似，ESP 头部域由 SPI 和序列号组成，尾部由填充域、填充长度域和“下一个头”组成。

ESP 隧道模式认证和加密服务所提供的安全性要强于 ESP 传输模式，因为在隧道模式

下，ESP对原始IP报文头部进行了加密和认证，而传输模式则没有。同时，由于ESP隧道模式对内部IP报头进行了加密，所以，当它运行在安全网关上时，它可以提供数据流保密服务。当然，由于加入了额外的IP报头，会产生额外的开销。

5.7.3 ESP协议处理及AH嵌套

如图5.24所示，ESP协议的处理过程类似于AH的处理过程，只是在数据包外出时，需要根据SA的参数加密特定流的IP报文，然后计算数据包的ICV(如果选择了ESP的数据源鉴别功能)；对于进入的ESP数据包，协议首先根据ICV进行消息完整性验证，然后，在正确解密该报文后，交由上层协议处理。

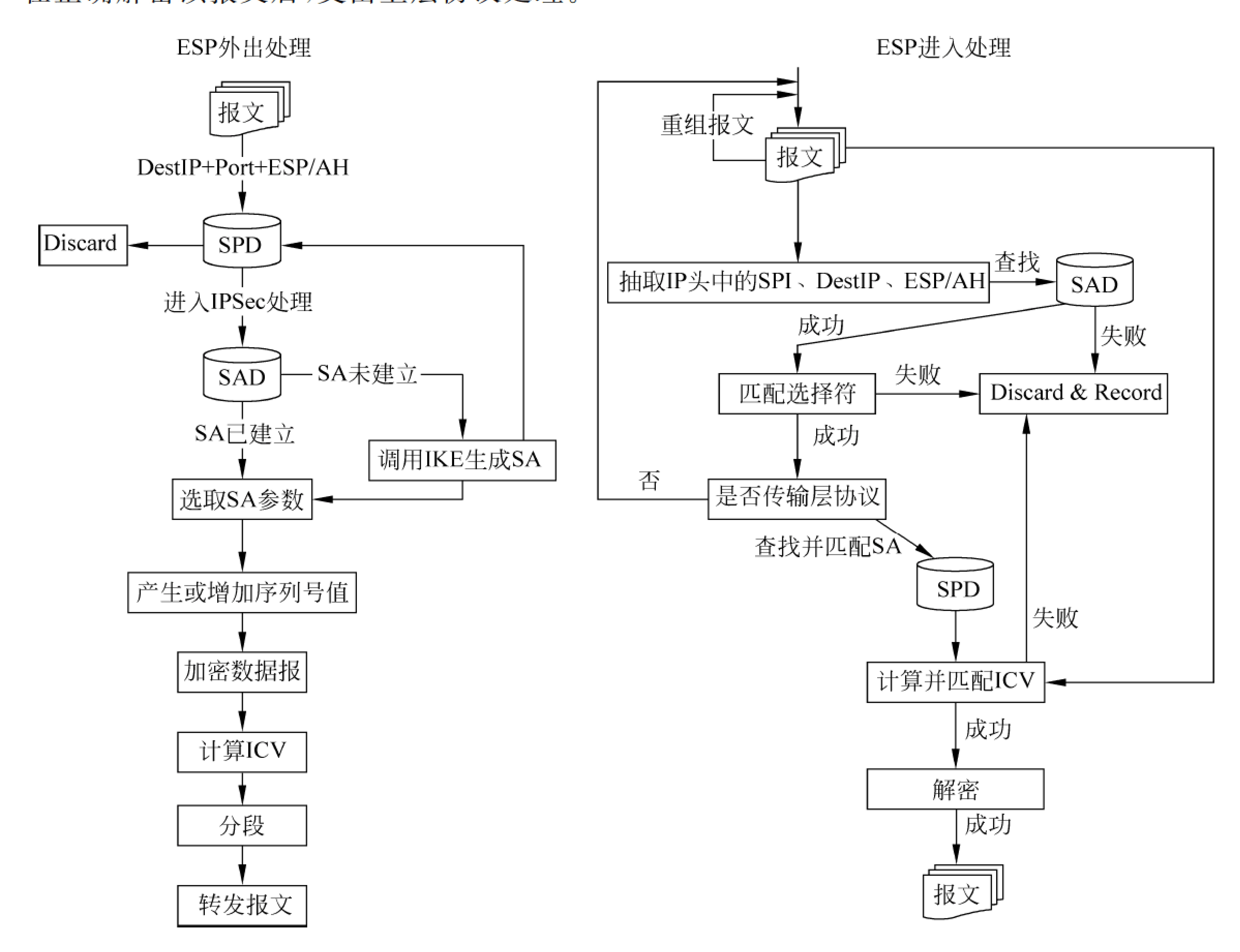


图5.24 ESP输出-输入处理流程

AH可以和ESP一起使用，其中传输模式下的数据包封装如图5.25所示，隧道模式下的数据包封装如图5.26所示。两种模式下，AH均被置于ESP报头之前，并对整个IP报文进行鉴别。图中的ESP只进行加密，没有进行数据源鉴别。

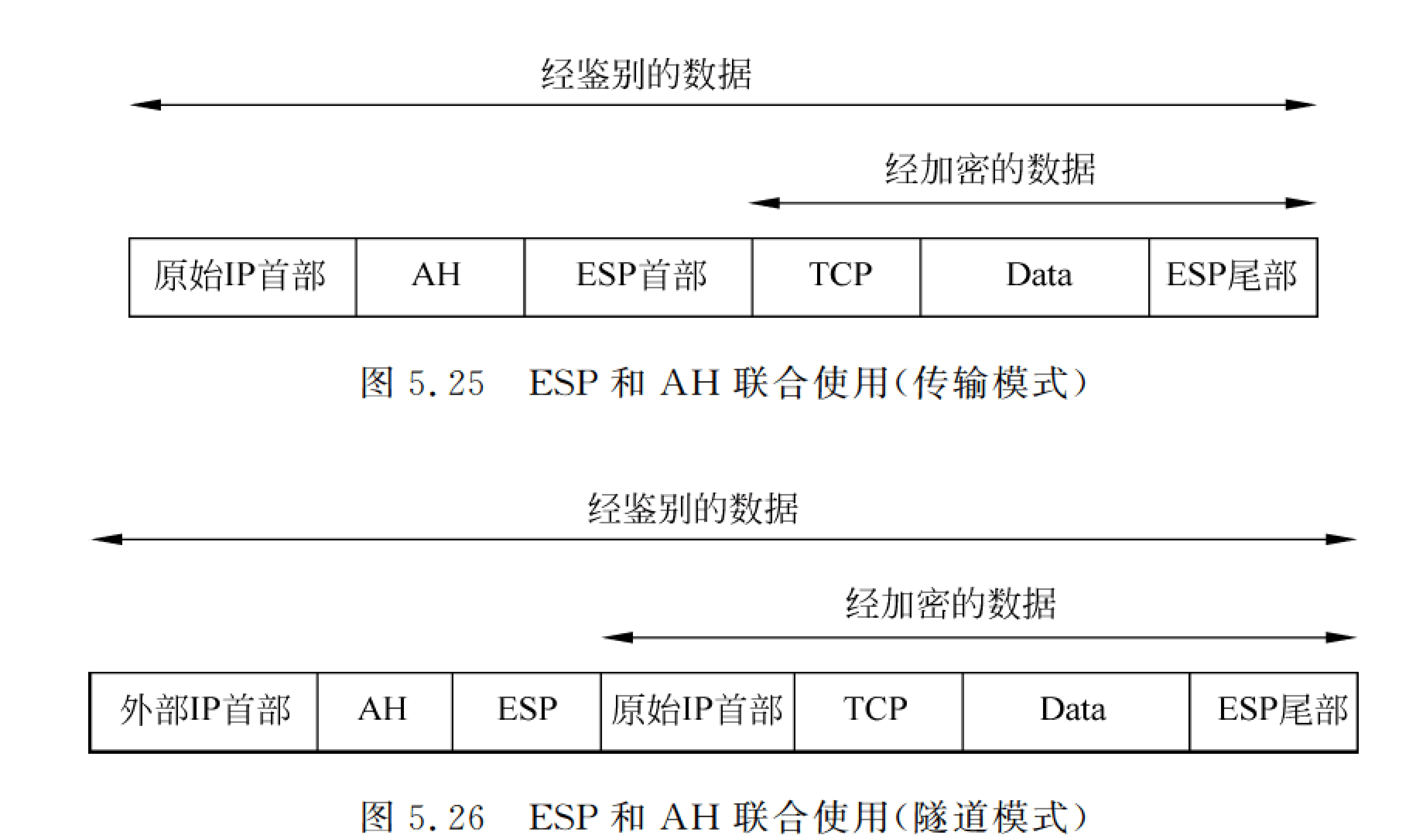


图 5.25 ESP 和 AH 联合使用(传输模式)

图 5.26 ESP 和 AH 联合使用(隧道模式)

5.8 IPSec 的应用

IPSec 可用于多种环境下增强网络的安全性,例如用于 IPSec VPN、端到端加密以及移动 IP 安全等。

1. IPSec VPN

VPN 通过公共 Internet 建立私有数据传输通道,将远程分支办公机构、商业伙伴和移动办公人员等互联起来,提供安全的端到端通信服务。VPN 兼备了公共网络和专用网络的优点,将公共网络丰富、灵活的功能与专用网络安全可靠的性能结合在一起。VPN 的实现应该满足安全通信的功能,因此可以采用某种形式的隧道机制对数据包进行加密封装。

IPSec 可以作为 VPN 隧道协议为 Internet 上的私有通信提供安全保护,它能够提供全面的安全服务,未来的 VPN 方案将会更多地利用 IPSec 实现。如图 5.27 所示,两个分支机构的安全网关之间通过运行 IPSec 隧道模式可以实现 IPSec VPN。

图 5.27 中,如果分支机构的内部网络是可信的,但 Internet 不可信,那么 IPSec 在每一个分支机构的安全网关可以通过 AH/ESP 的隧道模式为分支机构之间的通信提供安全服务,实现 IPSec VPN。这样,分支机构 A 发往分支机构 C 的数据包通过 IPSec 建立的隧道进行传输时,可以受到 IPSec 的安全保护,从而为公共网络上的通信提供安全服务。如果是 AH 隧道模式,整个数据包都会受到完整性验证保护,如果是 ESP 隧道模式,上层协议数据载荷会受到完整性保护,同时也会得到数据机密性保护。

如果分支机构内部的网络也不可信,而且数据流要求更高的安全级别,则可以在需要保

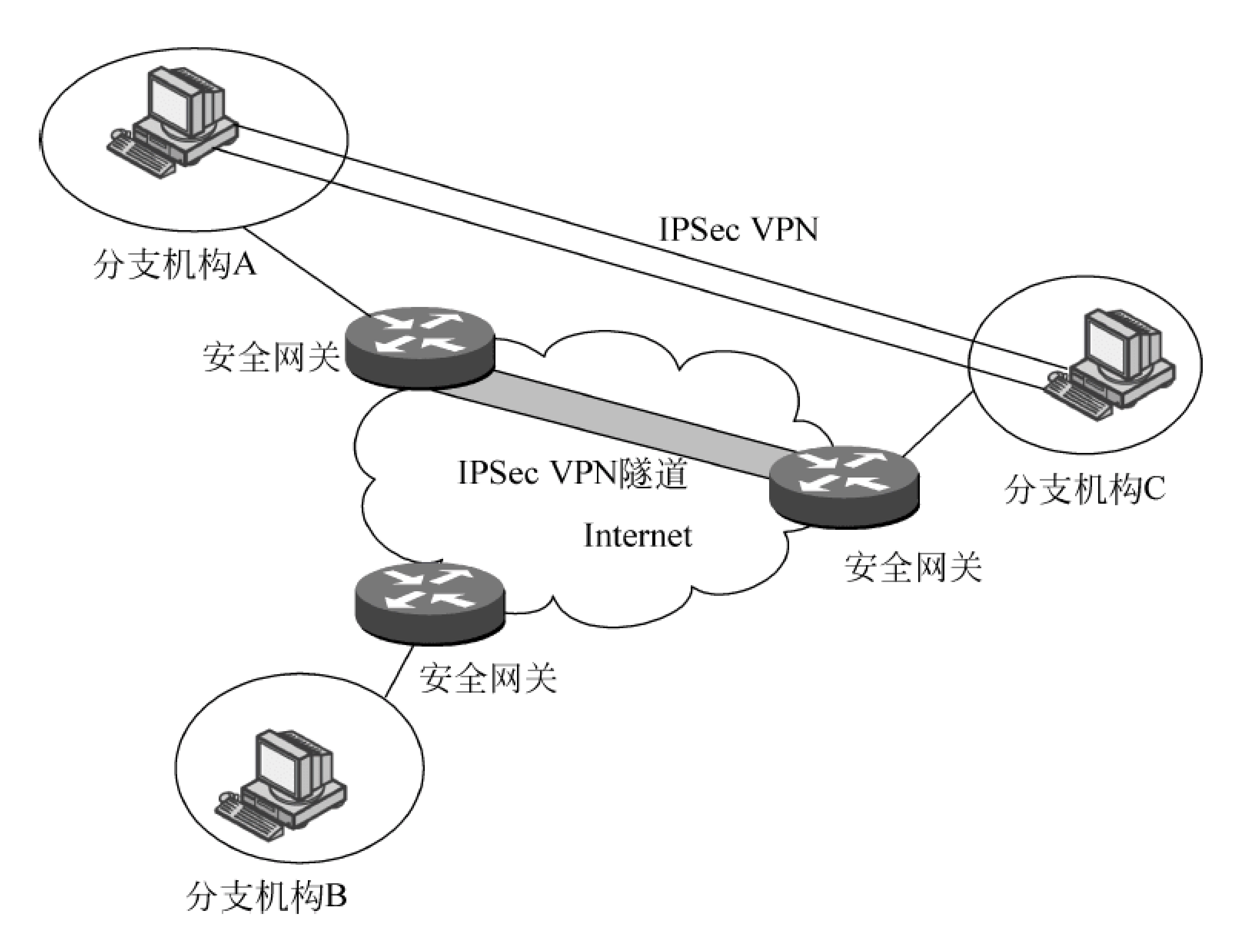


图 5.27 利用 IPSec 为分支机构提供 VPN 服务

护的主机之间的通信链路上使用端到端的认证和加密,方法是在主机上启用 IPSec,以提供安全服务。主机 A 与主机 C 之间的端到端的 IPSec 安全隧道为两个主机之间的通信提供安全服务,可以与安全网关的 IPSec 保护形成嵌套隧道。

IPSec VPN 可以通过防火墙、路由器和远程访问服务器等多种设备实现。

2. 端到端加密

操作系统中一般实现了 IPSec 协议栈,因此可以利用 IPSec 实现端到端加密,以保护 IP 及其上层协议的数据包。

此外,利用网络适配器及硬件设备也可实现 IPSec 的功能,此时这些设备能够直接提供 IPSec 加密和认证能力:一个网络适配器从物理链路上接收到 IPSec 数据包后,可以直接通过提取数据包头部的 SPI、源及目标地址以及安全通信协议,查找对应的 IPSec 安全协定,确定对数据包进行验证和解密所需的密钥,然后网络适配器可以直接在硬件中对数据包进行解密和验证,而不需要上层协议处理,因此可以提高协议的处理效率。

3. 移动 IP 安全

移动 IP 本身有一定的安全保障措施,但不够灵活方便。利用 IPSec 隧道可以增强移动 IP 的安全性。可以在移动 IP 网络中建立多种隧道,如:移动节点-家乡代理隧道、家乡代理-外地代理隧道以及外地代理-移动节点隧道等。

例如家乡代理-外地代理隧道(HA-FA 隧道)是从家乡代理到外地代理之间的安全通道。如图 5.28 所示,应用这种隧道可以为已经存在的移动 IP 隧道增加 IPSec 保护。这种

隧道也可以支持包括数据完整性的数据源验证和数据机密性。可以使用这种隧道来建立移动节点所属家乡子网和移动节点当前访问的外地子网之间的虚拟专用网。

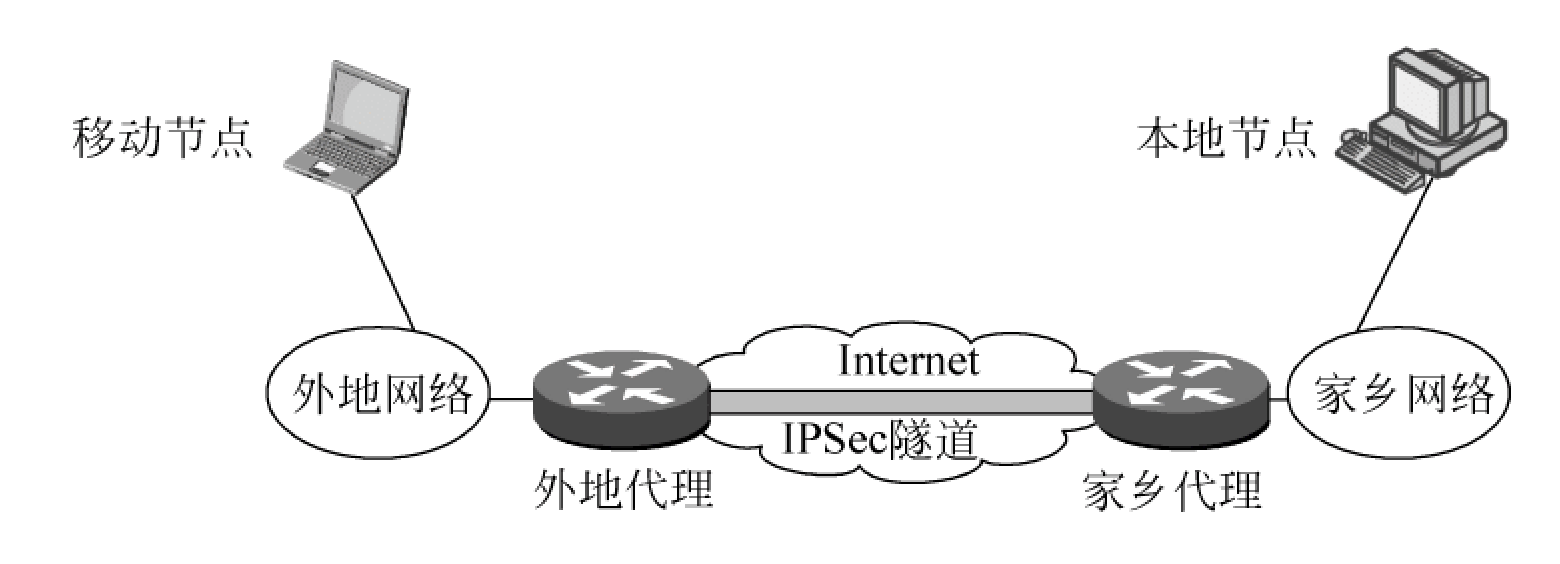


图 5.28 HA-FA 隧道

这种家乡代理-外地代理隧道可以抵御开放的 Internet 上的被动和主动攻击,因为只有移动节点满足了外地代理的认证和系统安全要求时,隧道才可以允许移动节点访问它的家乡子网。

本章实验

1. 利用 Windows 的 IPSec 协议栈实现安全的网络层通信。
2. 使用 ETHEREAL 工具软件分析加密前和 IPSec 加密后的 IP 数据包。

思考题

1. IPSec 中安全策略和安全协定 SA 的关系是什么? SA 中的内容如何产生?
2. 结合实验理解 IPSec 中 IKE 的作用。
3. 考虑如下网络环境:一台主机和另一台主机通过 Internet 进行通信,并且双方采用 IPSec 传输模式进行安全 IP 通信,源主机使用内部私有 IP 地址,由其本地网关进行网络地址转换,这时可能出现传输模式下 IPSec 协议无法正常工作的情况,试分析其原因。

第6章

传输层安全协议

6.1 SSL 协议

6.1.1 SSL 概述

安全套接层(SSL)协议提供传输协议之上的可靠的端到端安全服务,为两个通信对等实体之间提供机密性、完整性以及鉴别服务。

SSL 最初是由 Netscape 公司于 1994 年开发的。SSL 历经多次修订,其中,版本 1 未得到广泛使用。版本 2 即 SSLv2 于 1995 年发布,是为了满足 Web 通信的需要而设计的。SSLv3 也在 1995 年发布,SSLv3 对 SSLv2 中的多处安全问题进行了修正。为了设计一种更加强壮而且更易分析的系统,SSLv3 需要支持更多的算法。最终,这些要求使 SSLv3 与 SSLv2 完全不同,只保留了一些基本的协议特色。

1996 年 IETF 的 TLS 开发完成,TLSv1 是基于 SSLv3 的。并且,Netscape、Microsoft 都支持 TLSv1。

SSL 提供 4 个基本功能,即身份认证、数据加密、消息完整性保护和密钥交换。采用两种加密技术,即非对称加密和对称加密。非对称加密包括认证和交换加密密钥,对称加密则使用会话密钥加密数据传输。

如图 6.1 所示,SSL 是独立于应用层协议的,位于传输层协议和应用层协议之间。默认情形下,SSL 使用 TCP 端口 443 进行通信。SSL 常用于 HTTP 协议,但也可用于其他应用层协议,如 FTP、SMTP 和 Telnet 等。

SSL 的体系结构如图 6.2 所示,包括 SSL 握手协议,SSL 记录集协议,更改密码规约和告警协议。

1. SSL 握手协议(SSL handshake protocol)

SSL 握手协议中,客户端和服务器之间基于非对称加密算法相互鉴别,协商加密算法

HTTP	FTP	SMTP…
SSL/TLS		
TCP		
IP		

图 6.1　SSL 协议层次

和基于对称加密算法的会话密钥，提供连接安全性。连接安全性包括身份鉴别，至少对一方实现鉴别，也可以是双向鉴别；协商得到的共享密钥是安全的，中间人不能知道；协商过程是可靠的。SSLv3 提供对 Deffie-Hellman 密钥交换算法、基于 RSA 等密钥交换机制的支持。

2. SSL 记录集协议(SSL record protocol)

SSL 记录集协议建立在可靠的传输协议(如 TCP)基础上，用来封装高层协议，提供连接安全性。使用对称加密算法保证交互信息的机密性，用来对数据进行认证和加密的密钥通过 SSL 的握手协议来协商。使用 HMAC 进行完整性验证保证了交互信息的完整性。SSLv3 对数据认证(完整性验证)支持 MD5 和 SHA 算法，数据加密支持 R4 和 DES 等算法。

3. 更改密码规约协议(SSL change cipher spec protocol)

更改密码规约协议是最简单的特定于 SSL 的协议，它存在的目的是为了标识密码策略的变化。在完成握手协议之前，客户端和服务器都要发送这一消息，以便通知对方其后的记录将使用协商的密码规约以及相关联的密钥来保护。更改密码规约协议消息是在握手过程中安全参数协商好之后发送的，要在握手协议的 Finished 消息发送之前发送。所有意外的更改密码规范消息都将生成一个“意外消息”(Unexpected_message)进行告警。

4. 告警协议(SSL alert protocol)

SSL 记录层支持的内容类型之一是告警类型。告警协议将告警消息及其严重程度传递给 SSL 会话中的主体。告警消息使用当前连接状态所指定的方式来压缩和加密。当任何一方检测到一个错误时，检测的一方就向另一方发送一个告警消息。如果告警消息有一个致命的后果，则通信的双方应立即关闭连接。双方都需要舍弃任何与该失败的连接相关联的会话标识符、密钥等。对于所有的非致命的错误，双方可以缓存信息以恢复该连接。

SSL 握手协议	更改密码规约协议	告警协议	HTTP
SSL 记录集协议			
TCP			
IP			

图 6.2　SSL 协议构成

6.1.2 SSL 连接与会话

SSL 中有两个重要的概念，一个是 SSL 连接(connection)，连接是能提供合适服务类型的传输(在 OSI 分层模型中的定义)，对 SSL 这样的连接是对等关系，连接是暂时的，每个连接都和一个会话相关。另一个概念是会话(session)，SSL 会话是客户机和服务器之间的 SSL 协议对等实体之间的关联(一次会话)，会话由握手协议创建，它定义了一组可以被多个连接共用的密码安全参数。会话和连接的关系如图 6.3 所示。对于每个连接，可以利用会话来避免对新的安全参数进行代价昂贵的协商。

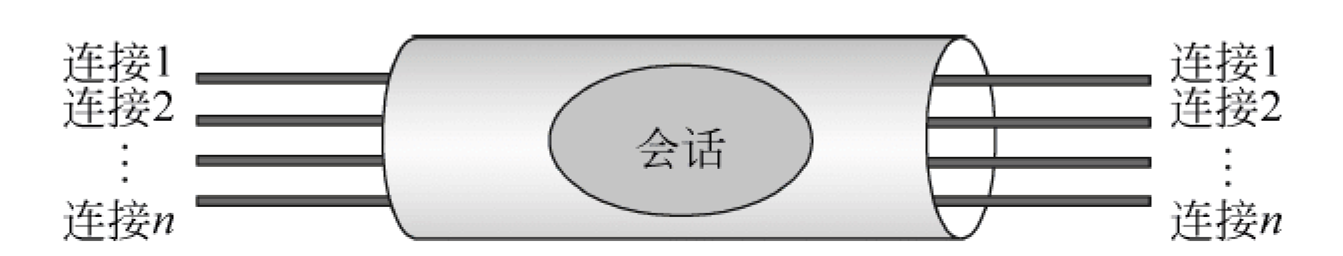


图 6.3 SSL 的连接与会话

SSL 的每个会话和连接都包含一些参数，会话和连接的参数分别如表 6.1 和表 6.2 所示：

表 6.1 会话状态参数

会话标识符 (session identifier)	由服务器选择的任意字节，用来确定活动或可恢复的会话状态
对等实体证书 (peer certificate)	对等实体的 X.509v3 证书。该状态的元素可为空
压缩方法 (compression method)	压缩数据的算法优于加密算法
加密规格(cipher spec)	指定批量数据加密算法和用于 MAC 计算的哈希算法。同时指定密码属性
主控密钥(master_secret)	由客户机和服务器共享的 48 位密码
是否可恢复(is resumable)	用来确定会话是否可用于初始化新连接的标志

表 6.2 连接状态参数

随机字	每个连接中，服务器和客户机选择的字节序列
服务器写 MAC 密钥	Server_write_MAC_secret，用于对服务器发送的数据计算 MAC 使用的密钥
客户机写 MAC 密钥	Client_write_MAC_secret，用于对客户机发送数据计算 MAC 使用的密钥
服务器写密钥	Server_write_secret，用于服务器对数据加密和客户端对数据解密的对称密钥
客户机写密钥	Client_write_secret，用于客户机对数据加密和服务器对数据解密的对称密钥
初始化向量 IV	当使用 CBC 模式的分组密文时，为每个密钥维护的初始化向量。该字段首先被 SSL 握手协议初始化，然后每个记录集最终的密文块被保留下来作为下一个记录的 IV
序列号	每一方为每个连接的传输和接收报文维持着单独的序列号，当一方发送和接收更改密码规约报文时，相应的序列号被设置为 0

6.1.3　SSL 握手协议

SSL 协议中，客户端和服务器首先通过握手过程来获得密钥，此后在记录集协议中使用这个密钥来加密客户端和服务器间的通信信息。握手过程首先采用非对称加密的方法来交换信息，使得服务器获得客户端提供的对称加密的密钥(pre_master_secret)，然后服务器和客户端使用这个 pre_master_secret 来产生会话密钥。SSL 握手协议包括以下三个阶段。

- 建立安全能力。即客户端和服务器通过 hello 消息协商会话连接的参数。
- 客户端鉴别和密钥交换。即服务器向客户端发送服务器证书后，客户端验证服务器证书，通过验证后，产生 pre_master_secret，并对 pre_master_secret 进行加密，然后把密文发送给服务器。随后，客户端和服务器均使用这个 pre_master_secret 来产生本次的会话密钥 master secret。
- 握手完成。即客户端和服务器都向对方发送 finish 消息，标志 SSL 握手完成。

下面分别描述以上三个阶段中，客户端和服务器间交互的具体步骤和使用的握手消息。

1. 建立安全能力

客户端和浏览器通过相互发送 hello 消息，统一 SSL 连接中的参数。这个过程如图 6.4 所示。客户端向服务器发送 Client Hello 消息，服务器向客户端发送 Server Hello 消息。

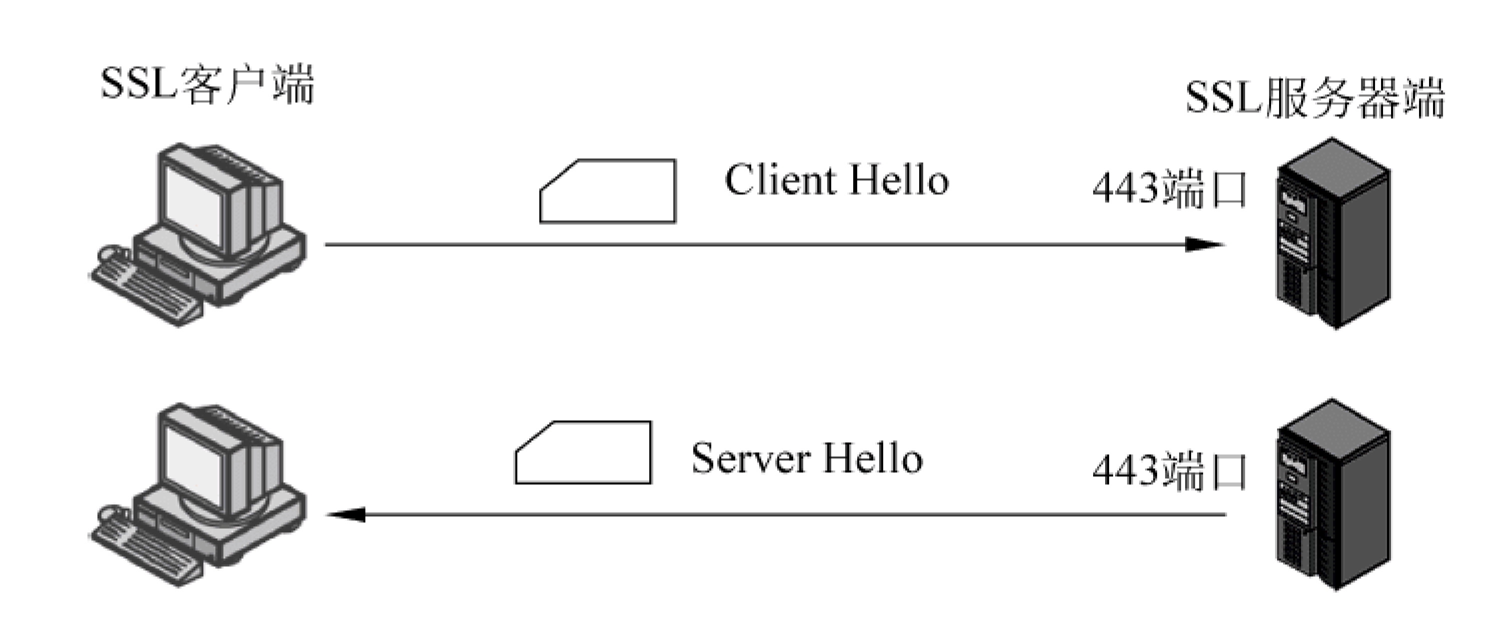


图 6.4　建立安全能力

其中，客户端给服务器发送的 hello 消息的构成如表 6.3 所示。

2. 客户端鉴别和密钥交换

(1) 鉴别消息交换

这一步是整个握手过程的核心。在上一步服务器向客户端回复了 hello 消息后，服务器需要给客户端发送服务器证书，客户端收到证书后执行服务器验证。服务器证书是服务器从 CA 处获得的，用于证明服务器的身份。这一步，服务器可能给客户端发送如下 4 条消息，如图 6.5 所示。

表 6.3 客户端 hello 消息的组成

消息内容	描述
SSL 版本号(SSL version)	客户端能理解的最高版本号,通常是 TLSv1 或 SSLv3
密码交换(key exchange)	用于识别交换密钥的算法,可使用的密钥交换算法包括 RSA、Diffie-Hellman 和 Fortezza 等
数据加密(data encryption)	用于识别客户端所能支持的加密算法,可使用的数据加密算法包括 RC2-40、RC4-128、DES、DES 40、3DES、IDEA 和 Fortezza
信息摘要(message digest)	用于保证消息的完整性,通常使用 SHA 或者 MD5 算法来计算哈希值
数据压缩方法(data compression)	用于信息交换的压缩方法,可使用 PKZip 或 gzip
随机字	用于计算密钥而产生的随机数

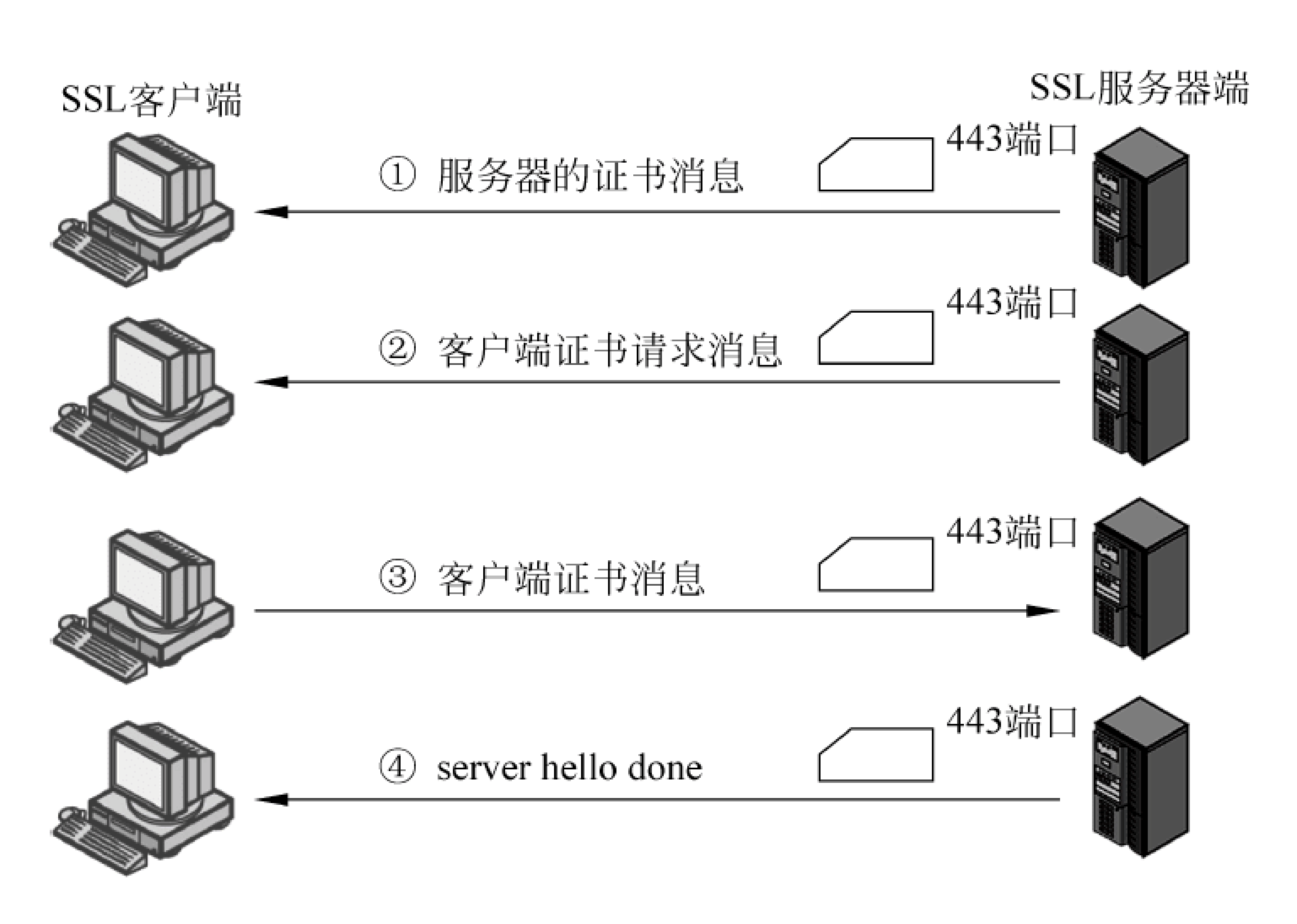


图 6.5 鉴别消息交换过程

① 服务器的证书消息(server certificate)。其中包括服务器的标识符、CA 私钥签名、服务器公钥等信息,如图 6.6 所示。

② 客户端证书请求消息(client certificate request)。如果配置服务器的 SSL 需要验证客户端身份,就会发出请求要求客户端提供用户证书(client certificate request);客户端证书请求消息包括识别公共密钥的证书类型,服务器能够接受的是一系列证书认证机构的名字。

③ 客户端证书消息(client certificate)。客户端向服务器发送其公钥证书。

④ server hello done 消息。服务器向客户端发送该消息,表示握手过程结束,该消息没

有参数。

（2）证书验证过程

在客户端收到服务器证书后，首先需要通过验证服务器证书来验证服务器是否可信任。以下描述客户端验证服务器证书的详细过程，如图 6.6 所示。

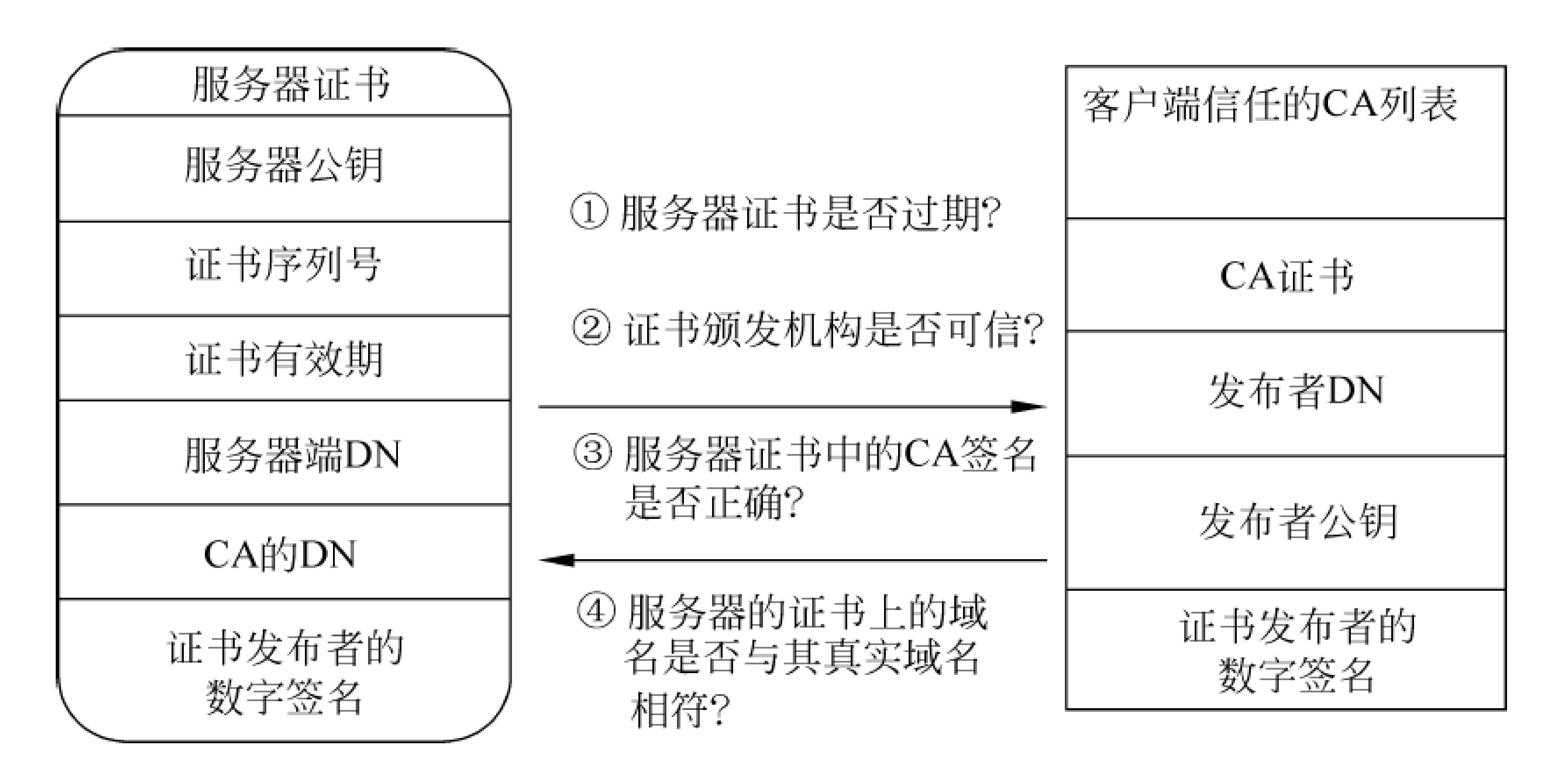


图 6.6　客户端验证服务器证书

① 客户端检查服务器端证书是否过期。如果证书过期，验证程序中止；否则，进入第二步。

② 验证证书颁发机构是否可信。每个使用 SSL 的客户端都有一组可信 CA。根据这个可信 CA 列表，客户端进一步判断服务器是否可信。若证书颁发机构的标识名(distinguished name，DN)与客户端的可信列表相符，则执行第三步。

③ 验证服务器证书中的 CA 签名。客户端使用 CA 的公钥(CA 的公钥可以从客户端已存的可信 CA 列表中获得)来验证它的数字签名的真伪，即对服务器证书中的 CA 的私钥签名进行验证。若服务器证书内容上有改变或签名验证失败，客户端就不能通过对服务器端的身份鉴别，验证程序终止。如果 CA 的签名为真就转到第四步。

④ 验证服务器的证书上的域名是否与其真实域名相符。

这一步用来确定服务器真正位于它所在的网络地址。从技术角度讲，虽然第四步不属于 SSL 协议，但它可抵御类似“中间人攻击”的网络威胁。客户端执行这一步可以拒绝在服务器或域名不符时建立连接。如果相符，则服务器验证完成。

通过了服务器验证后，客户端和服务器开始进行密钥协商。

（3）密钥交换

服务器证书通过客户端验证后，进入密钥协商阶段，通信双方按如下步骤执行协议。

① 客户端产生一系列伪随机数作为 pre_master_secret。这个 pre_master_secret 将用来产生服务器和客户端之间的会话密钥。

② 客户端使用服务器证书中的服务器公钥加密 pre_master_secret，并且把这个加密后的

pre_master_secret 通过 Client Key Exchange 消息发送给服务器端(如图 6.8 和表 6.4 所示)。

③ 服务器和客户端使用事先协商好的算法,以 pre_master_secret 作为输入参数之一,进行密钥计算,产生加密使用的各种密钥对。

在第三步中,客户端和服务器均使用 pre_master_secret 产生三对密钥(对称密钥),这三对密钥用于服务器和客户端加密和鉴别通信数据时使用。这三对密钥分别如下。

- Client_write_secret 和 Server_write_secret。其中,Client_write_secret 是客户端对数据加密以及服务器对数据解密时所用的密钥,而 Server_write_secret 是服务器对数据加密以及客户端对数据解密时所用的密钥。
- Client_write_MAC_secret 和 Server_write_MAC_secret。用于客户端和服务器计算消息摘要时使用。其中,Client_write_MAC_secret 用于客户端计算消息摘要和服务器对数据完整性进行验证,Server_write_MAC_secret 是服务器计算消息摘要以及客户端进行数据完整性验证时所用的密钥。
- Client write IV 和 Server write IV。用来计算初始化向量 IV(Initialization Vector)。

客户端和服务器端都基于 pre_master_secret 和 hello 消息中协商好的加密算法生成这三对密钥。

SSL 握手过程中密钥产生的关系如图 6.7 所示。

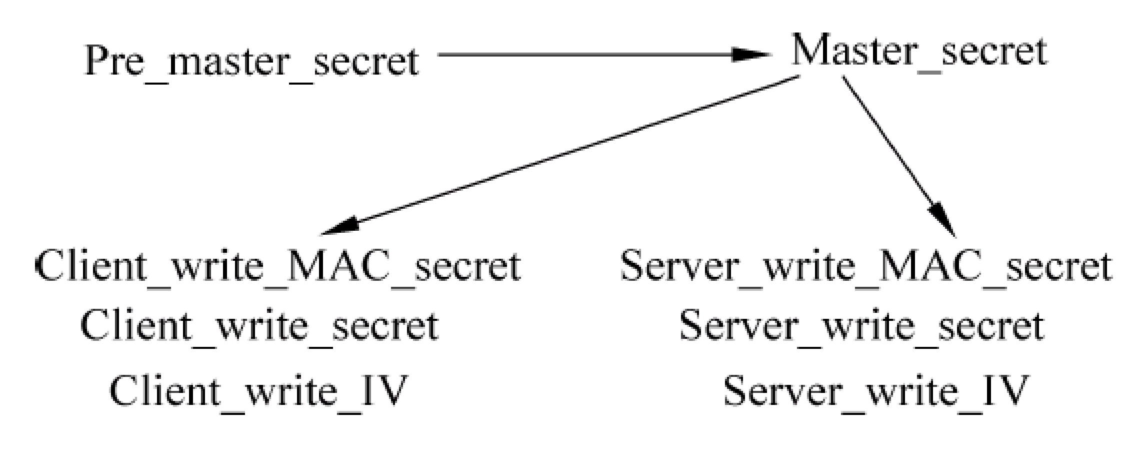


图 6.7 各种密钥

3. 握手完成

客户端和服务器端都产生了用于会话的对称密钥后,他们通过发送 finished 消息标志握手过程结束。如图 6.8 所示,客户端和服务器都先后向对方发送 finished 消息,该消息中,客户端使用刚产生的共享密钥(client_write_secret)加密消息来鉴别对方。这个消息使得双方都确认安全的连接已经建立。

以上描述的整个握手过程中的各种消息如图 6.8 所示。

可以看到,客户端和服务器通过 hello 消息统一 SSL 中的一些参数,然后服务器端向客户端发送服务器证书。客户端对服务器证书的验证通过后,产生并加密发送 pre_master_secret 到服务器端,然后双方都使用这个 pre_master_secret 产生各种会话过程中用于加密交互信息的对称密钥。

握手消息的格式如图 6.9 所示。

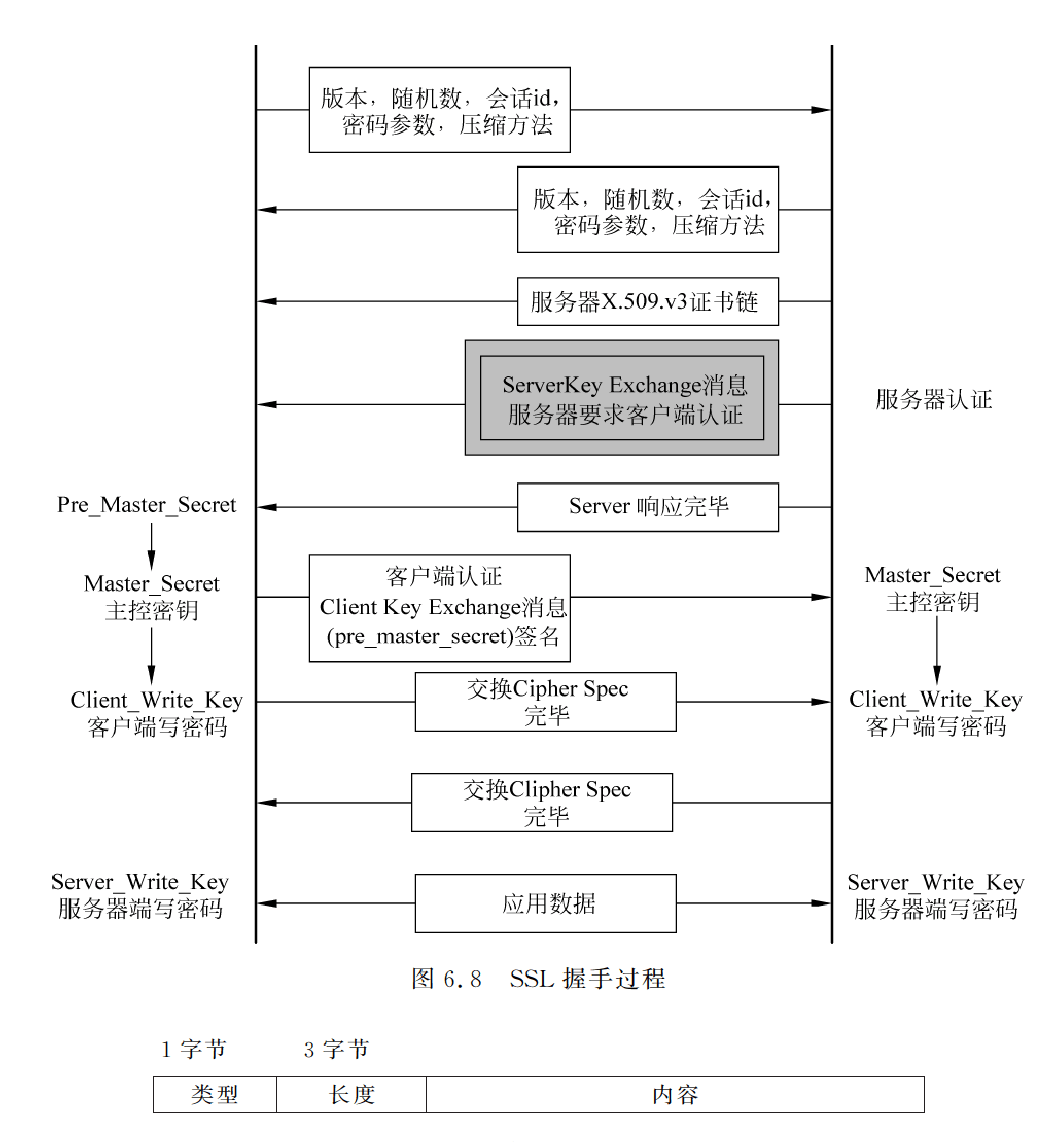


图 6.8 SSL 握手过程

图 6.9 SSL 握手协议报文格式

表 6.4 中列出了消息类型及其参数和描述。

表 6.4 SSL 握手协议报文消息描述

消息	参数	描述
Hello_request	Null	启动握手协议
Client_hello	版本、随机数、会话 ID、密码参数、压缩方法	启动 SSL 会话，消息中标识密码和压缩方法列表
Server_hello		
Server certificate	X.509v3 证书链	服务器向客户端发送请求验证的消息
Server_key_exchange	参数，签名	密钥交换
Certificate_request	类型，CAs	服务器要求对客户端认证

续表

消　息	参　数	描　述
Server_done	Null	指示服务器的 hello 消息发送完毕
Certificate_verify	签名	对客户证书进行验证
Client_key_exchange	pre_master_secret，签名	密钥交换
finished	Hash 值	标志密钥交换和鉴别过程成功

6.1.4　SSL 记录集协议

SSL 记录集协议提供的服务包括消息的机密性和完整性等。首先，由于握手协议定义了共享的可用于对 SSL 有效载荷进行常规加密的密钥及初始/后续的 IV，对压缩的数据和数据摘要进行了加密，客户端和服务器在通信过程中的数据都是密文，这就保证了数据的机密性；另外，通过握手协议定义了共享的、可用于生成报文(migration authorisation code，MAC)的密钥，对压缩数据计算了消息摘要。接收方在收到数据后，使用共享密钥进行密文解密后，然后使用它的 MAC 密钥计算压缩数据的消息摘要 MAC 值，将其和附加在 SSL 记录集中的 MAC 值进行比对，如果相等，则说明压缩数据在传输过程中没有被修改过，这就保证了数据的完整性。

在 SSL 协议中，所有的传输数据都被封装在记录集协议中。记录集报文由记录头和长度不为零的记录集数据组成。SSL 记录集协议包括对记录头和记录数据格式的规定。

通过 SSL 握手协议获得的会话密钥将用于加密服务器和客户端在记录集中交互的数据。如图 6.10 所示，记录集协议对原始数据的加密过程如下。

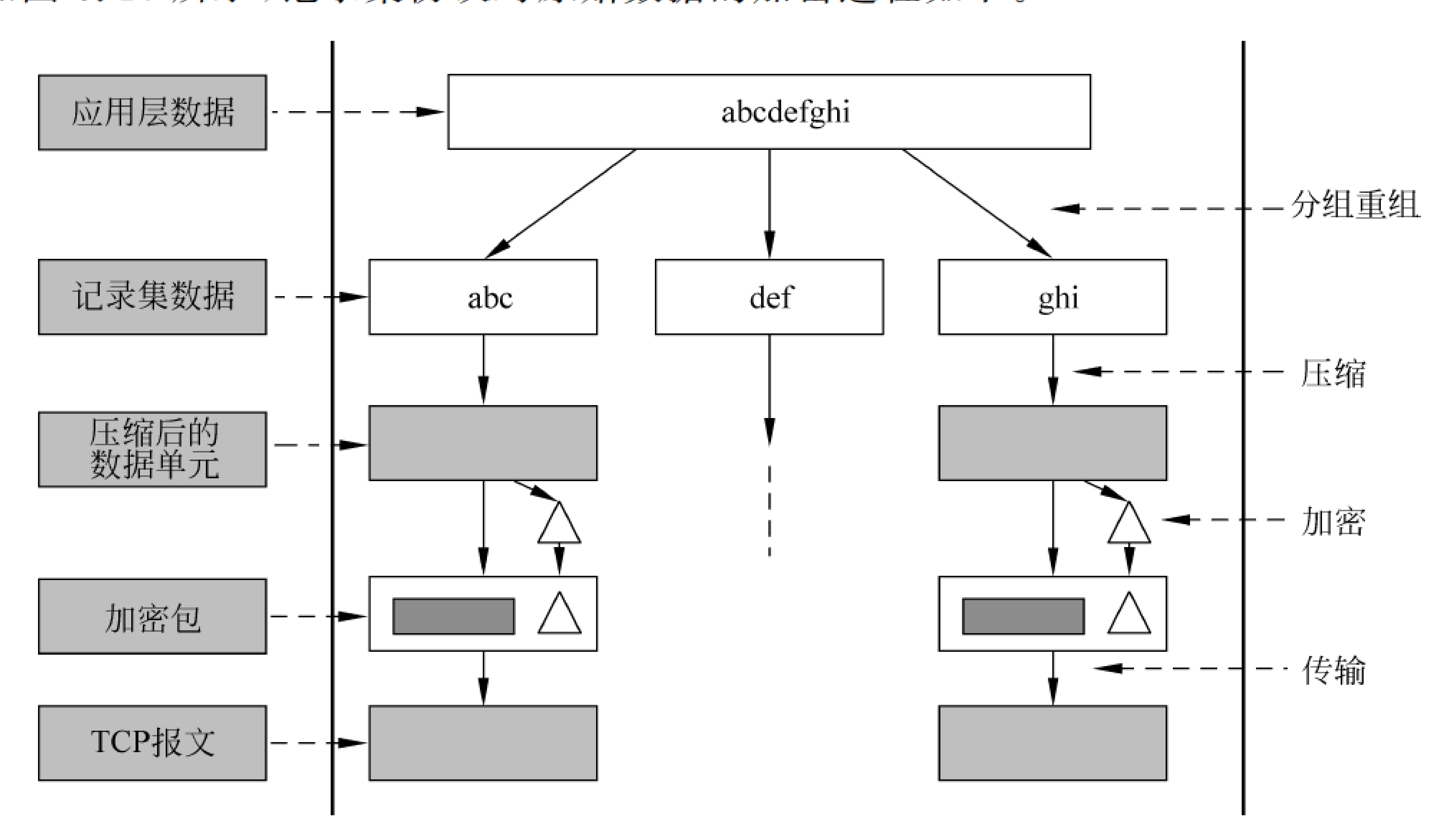


图 6.10　对数据进行加密

① 把来自上层协议的数据进行分组。

② 对每个分组进行压缩，形成压缩数据。

③ 发送端根据压缩数据和其 MAC secret 密钥计算压缩数据的消息摘要。

④ 发送端把发送的压缩数据和消息摘要一起使用其 Write secret 加密。

⑤ 在密文上增加 SSL 记录头。

通过以上过程即把原始数据加密为 SSL 协议的记录集，其中可选择的加密算法如表 6.5 所示。

表 6.5 SSL 记录集协议使用的加密算法

块密码算法	流密码算法	块密码算法	流密码算法
IDEA(128)	RC-40(40)	DES	
RC2-40(40)	RC4-128(128)	DES 3	
DES-40		Fortezza	

如果采用 CBC(cipher block chaining)模式加密，那么加密算法由会话状态中的 Cipher spec 参数决定。CBC 模式是指一个明文分组在被加密之前要与前一个的密文分组进行异或运算。当加密算法用于此模式的时候除密钥外，还需协商一个初始化向量(initialization vector，IV)，这个 IV 在第一次计算的时候需要用到。

图 6.11 给出了 SSL 记录集协议的报文格式，其中：

- 内容类型为 8 位，说明了上层协议类型。
- 主次版本号分别是 16 位。
- 压缩长度占 16 位，表示压缩后的数据字节数。
- 压缩明文和 MAC 消息摘要是被加密了的密文。

内容类型	主版本号	次版本号	压缩长度
被压缩(可选)的明文			
MAC(0,16 or 20 字节)			

图 6.11 SSL 记录集格式

6.1.5 SSL 密码计算

共享主密钥(master_secret)由客户机和服务器共享，是通过 pre_master_secret(也称作次密钥)生成的临时 48 字节值。

从 6.1.3 节的握手过程可知，master_secret 分两个步骤生成。首先通过握手过程交换 pre_master_secret，然后双方再根据 pre_master_secret 来计算 master_secret。

pre_master_secret 交换方法可以采用 RSA 或 Diffie-Hellman 算法：

- 在 RSA 算法中，由客户端生成 48 字节的次密钥，并用服务器的 RSA 公钥加密后发

送到服务器，服务器用其私钥解密，得到 pre_master_secret。

- 在 Diffie-Hellman 算法中，客户端和服务器同时生成 Diffie-Hellman 公钥，密钥交换后双方执行 Diffie-Hellman 计算，创建共享次密钥 pre_master_secret。

得到 pre_master_secret 后，交互双方分别按如下方法计算 master_secret。

```
master_secret =
      MD5(pre_master_secret + SHA('A' + pre_master_secret +
         ClientHello.random + ServerHello.random)) +
      MD5(pre_master_secret + SHA('BB' + pre_master_secret +
         ClientHello.random + ServerHello.random)) +
      MD5(pre_master_secret + SHA('CCC' + pre_master_secret +
         ClientHello.random + ServerHello.random));
```

其中，clientHello. random 和 ServerHello. random 是握手协议中 hello 消息中的两个临时交换值。master_secret 计算完毕后，服务器和客户端把各自的 pre_master_secret 删除。

计算得到 master_secret（主密钥）后，还需要创建其他密码参数。CipherSpec（密码参数）用于指定加密 SSL 记录集和验证 SSL 记录集完整性的所采用的算法。典型的方法是使用 DES 算法加密数据和使用 MD5 来计算消息摘要。

Master_secret 被哈希算法变换成一系列字节，然后被分配为 MAC secret（验证完整性的 MAC 密钥）、write secret（加密密钥）和 write IV 三类密钥。这些密钥是当前需要的密码参数（CipherSpec）。

这三类密钥的产生需要以 master_secret，ClentHello. random，ServerHello. random 为输入，产生的过程分为如下两步。

① 采用如下的公式计算 key_block，一直到输出的 key_block 满足要求。

```
key_block =
       MD5(master_secret + SHA('A' + master_secret +
           ServerHello.random +ClientHello.random)) +
       MD5(master_secret + SHA('BB' + master_secret +
           ServerHello.random + ClientHello.random)) +
       MD5(master_secret + SHA('CCC' + master_secret +
           ServerHello.random + ClientHello.random)) + [...];
```

② 获得 key_block 后，这个 key_block 被分解为各个密钥，如下所示。

- client_write_MAC_secret[CipherSpec. hash_size]
- server_write_MAC_secret[CipherSpec. hash_size]
- client_write_key[CipherSpec. key_material]
- server_write_key[CipherSpec. key_material]
- client_write_IV[CipherSpec. IV_size]
- server_write_IV[CipherSpec. IV_size]

6.1.6 SSL 协议的应用

1. HTTPS

SSL 现已成为网络用来鉴别网站和网页浏览者身份，以及在浏览器使用者和 Web 服务器之间进行鉴别和加密通信普遍使用的技术。例如，我们登录一些银行等安全要求比较高的网站时，经常可以看到 HTTPS 开头的 Web 地址，简单地说，它是 HTTP 的安全版本，即 HTTP Over SSL。HTTPS 使用 SSL 进行信息交换，使得所有的 HTTP 数据在传输过程中都是加密的。

当使用基于 SSL/TLS(通常使用 https:// URL)向站点进行 HTTP 请求时，从服务器向客户机发送一个证书。浏览器使用已安装的公共证书来验证服务器的身份。身份鉴别通过后，浏览器和服务器之间使用握手协议交换密钥。

2. SSL VPN

随着 Web 应用的增多以及远程接入需求的增长，SSL VPN 被广泛使用。虚拟专用网 VPN 主要应用于虚拟连接网络，它可以确保数据的机密性并且具有一定的访问控制功能。VPN 可以扩展企业的内部网络，允许企业的员工、客户以及合作伙伴利用 Internet 访问企业网，而成本远远低于传统的专线接入。过去，VPN 总是和 IPSec 联系在一起，因为它是 VPN 加密信息实际用到的协议。IPSec 运行于网络层，IPSec VPN 多用于连接两个网络或点到点之间的连接。

SSL VPN 指的是使用者利用浏览器内建的 SSL 包处理功能，用浏览器连接公司内部 SSL VPN 服务器，让使用者可以在远程计算机执行应用程序，读取公司内部服务器的数据。它采用标准的安全套接层 SSL 对传输中的数据包进行加密和鉴别，从而在应用层保护了数据的安全性。

SSL VPN 一般的实现方式是在企业的防火墙后放置一个 SSL 代理服务器。如果用户希望安全地连接到公司网络上，那么当用户在浏览器上输入一个 URL 后，连接首先被 SSL 代理服务器处理，验证用户的身份，然后 SSL 代理服务器将提供远程用户与各种不同应用服务器之间的连接。

高质量的 SSL VPN 解决方案可保证企业进行安全的全局访问。在不断扩展的互联网 Web 站点之间、远程办公室、传统交易大厅和客户端之间，SSL VPN 克服了 IPSec VPN 的不足，进一步保障访问安全，使得用户可以轻松实现安全易用、无须客户端安装且配置简单的远程访问。

3. Open SSL

OpenSSL 是一个开放源代码的、实现了 SSL 及相关加密技术的软件包，由加拿大的

Eric Yang 等发起编写。OpenSSL 是一个开源工具箱，在其上可以开发安全的套接字通信程序。使用 OpenSSL 的 API，可以实现消息摘要、文件的加密和解密、数字证书、数字签名等功能。此外，它还提供一个命令行工具。命令行工具可以完成与 API 同样的工作，而且更进一步，可以测试 SSL 服务器和客户机。

使用 OpenSSL API 进行安全编程，首先需要下载其源代码，也可以下载二进制包。OpenSSL 的官方网站为 http://www.openssl.org/，源代码可以从 ftp://ftp.openssl.org/source/上下载。此外，有的 Linux 发行版本中也附带了 OpenSSL 的二进制版本。

6.2 SSH 协议

6.2.1 SSH 概述

SSH(secure shell)是 IETF 的网络工作组制定的一族协议，其目的是要在非安全网络上提供安全的远程登录和其他安全网络服务。

类似于 SSL，SSH 也是建立在应用层和传输层基础上的安全协议。和 SSL 不同，SSH 主要解决的是密码在网络上明文传输的问题，因此通常用来替代 TELNET、FTP 等协议。

传统的 Telnet、FTP 和 Rlogin 等服务存在众多安全缺陷，例如使用弱密码单一认证机制；传输数据(包括账号和密码)为明文，容易被窃取、篡改和重放；这些服务的安全验证机制容易引发各种欺骗，比如中间人攻击等。为了克服这些安全缺陷，SSH 协议被设计出来。

SSH 使用多种加密方式和认证方式，解决了以上传统服务的数据加密、身份认证问题。SSH 成熟的公钥/私钥体系，为客户端和服务端之间的会话提供加密通道，解决了数据(包括密码)在网络上明文传输的不安全问题。SSH 还支持 CA、smart 卡等多种认证方式，解决了身份认证问题，可抵御重放攻击和中间人攻击。

SSH 的"加密通道"是通过端口转发实现的。可以在本地没有使用的端口和在远程服务器上运行的某个服务的端口之间建立"加密通道"。然后只要连接到本地端口，所有对本地端口的请求都被 SSH 加密并且转发到远程服务器的端口。

SSH 协议主要由传输层协议(the transport layer protocol)、用户认证协议(user authentication protocol)和连接协议(the connection protocol)组成，共同实现 SSH 的安全保密机制。在 SSH 的协议框架中，传输层协议提供服务器认证，数据机密性，消息完整性服务等的支持；用户认证协议为服务器提供对客户端的身份鉴别；连接协议将加密的信息隧道复用成若干个逻辑通道，提供给高层的应用层协议使用。各种应用层协议可独立于 SSH 基本体系之外，依靠这个基本框架，通过连接协议使用 SSH 的安全机制。

为了满足扩展性的要求，协议规范了所采用的密码算法、密钥协商方式和认证方式等的

命名规则，并统一协议中消息的格式。协议也允许在各个方向上充分协商加密、完整性、密钥交换、压缩及公钥算法和格式等。新的算法、扩展协议等可以自由地添加，只要它们符合协议规定的命名规则以及消息格式。

6.2.2 SSH 协议体系结构

SSH 协议从几个不同角度强化通信的完整性。如前所述，SSH 协议主要由 SSH 传输层协议、SSH 用户认证协议和 SSH 连接协议三个组件组成。每层提供不同类型的安全保护，并且可以与其他方式一起使用，其协议结构如图 6.12 所示。

SSH 应用层协议		
连接协议		
用户认证协议		
传输层协议		
服务器认证	数据机密性	数据完整性
TCP over IP		

图 6.12 SSH 协议构成

1. SSH 传输层协议

SSH 传输层协议提供数据通信加密处理、加密的主机身份认证、数据完整性校验以及数据压缩等多项安全服务。双方通信所需要的密钥交换方式、公钥密码算法、对称密钥密码算法、消息认证算法和哈希算法等都可以进行协商。传输层协议的主要目标是实现两台主机间认证时和认证后安全和保密的通信，通常运行于 TCP/IP 之上。

SSH 传输层协议中的认证是基于主机的，并不涉及客户端用户的身份认证，传输层产生了密码认证和其他服务需要的秘密数据。用户认证可以通过基于该协议之上、单独设计的协议来完成。这样，既保证了通信的安全性，又提供了协议的灵活性和扩展性。

2. SSH 用户认证协议

在传输层构建了一个安全隧道以在两个系统间传送信息后，服务器将告诉客户机它所支持的认证算法，客户机将用服务器支持的算法向服务器证明自己的身份。认证由服务器主导，客户端可以根据服务器提供的方法进行选择，这样一方面使服务器对认证有完全的控制权，同时也给客户端足够的灵活度。主要包括以下几种用户认证方式。

- 公钥认证方式：是 SSH 协议唯一要求的必须提供的认证方法。在这种方式中，用户用私钥来表明自己的身份。简单说，就是用户向服务器发送一个用自己私钥处理过的数字签名，服务器首先检查该用户的私钥是否可以作为一个有效的认证凭证（通过检查本地数据库中是否存有与之对应的公钥），然后检查该签名的有效性，如果两个条件都满足，用户的认证请求就可以被接受，否则认证失败。
- 密码认证：所有的应用都应该支持由服务器确定怎样编译密码及根据密码数据库验证密码。在此过程中，客户机和服务器都应该检验传输层所提供的机密性。如果没有提供加密，密码认证不能完成；如果没有机密性或 MAC，则不能改变密码。

- 基于主机的认证：根据用户来自的主机及远端主机上的用户名来认证。这种方式不适用于安全性要求较高的站点。另外，可以混合使用几种不同认证特征的用户认证方法。由服务器的本地策略决定使用哪一种方式（或混合方式）。采用多种认证时，认证强度取决于最弱的认证方式。

3. SSH 连接层协议

SSH 连接层协议主要的功能是完成用户请求的各种网路服务，而这些服务的安全性是由底层的 SSH 传输层协议和用户认证层协议实现的。在 SSH 传输层成功认证后，多个信道通过复用到两个系统间的单个连接上而打开。每个信道处理不同的终端会话。

客户基于服务器可以建立新的信道，每个信道在每一端被编排给不同的号码。在一方试图打开一个新的信道时，该信道在该端的号码随请求一起传送，并被对方存储，以用于指示特定类型业务的通信给该信道。这样可以使不同类型的会话不会彼此影响，而在关闭信道时也不会影响系统间建立的初始 SSH 连接。

6.2.3 SSH 协议分析

SSH 协议的主要目的是提高网络应用的安全性，以容易部署和使用为原则来实现这个目标，而以牺牲一些绝对安全作为代价。其特点如下。

- 所有的数据加密、身份认证，完整性校验及公钥算法均采用一些相对成熟、经过时间检验且被人们广泛使用的算法。
- 所有的算法都选用目前普遍认为安全的密钥长度，在相当长的一段时期内能够抵御密码分析。
- 所有的算法都是可以协商的，这样即使某种算法被攻破，也可以很容易地切换到另外一种算法而不用更改整个基础协议。

1. 消息及消息编号

因为引入了新的协议包头、填充项以及完整性校验码，所以整个数据包的长度被加长了。这种增加量对于大尺寸的数据包来说几乎可以忽略，但是对于像 Telnet 那样的单字节交互会话来说，这种开销就显得很不合适。

在连接层建立起来的逻辑信道中，每个信道上允许的最大数据包长度也是可以协商的。SSH 数据包格式如图 6.13 所示。

包长度 packet_length	填充域长度 padding_length	有效负荷 payload	随机填 random padding	MAC

图 6.13 SSH 协议数据包格式

SSH 协议中包括如下字段。

(1) 包长度。不包括消息认证代码和本身。

(2) 填充域长度。

(3) 有效负荷。如果压缩选项被选中，该部分将被压缩处理；长度为包长度－填充域长度－1。

(4) 随机填充域。用于保证包长度、填充域长度、有效负荷及填充域的总和为密码分组长度或者 8 的整数倍(取较大的数值)。

(5) 消息认证代码 MAC。如果双方协商了消息认证，该部分就是消息认证码。

SSH 的消息编号从 1～ 255，分配如下。

① 传输层协议。

- 1～19：传输层常用消息(如断开、忽略和调试等)。
- 20～29：算法协商消息。
- 30～49：与密钥交换方法相关的消息(不同的认证方式可能会重用)。

② 用户认证协议。

- 50～59：用户认证常用消息。
- 60～79：与用户认证方法相关的消息(不同的认证方法可能会重用)。

③ 连接协议。

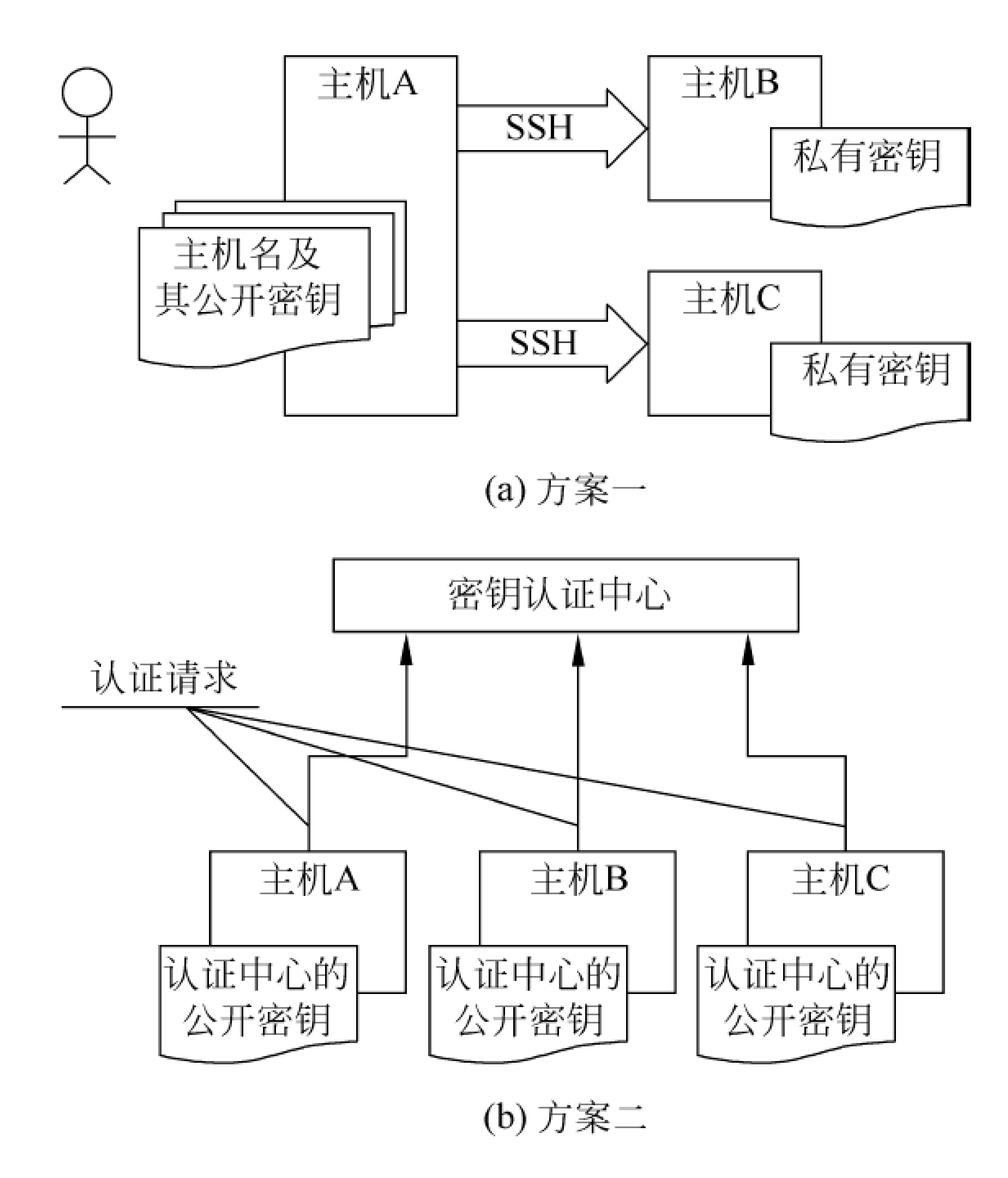
- 80～89：连接层常用消息。
- 90～127：信道相关消息。
- 128～191：为客户协议保留。
- 192～255：本地扩展用。

2. 主机密钥机制

对于 SSH 这样以提供安全通信为目标的协议，一套完备的密钥机制必不可少。由于 SSH 协议是面向互联网中主机之间的互访与信息交换的，因此主机密钥成为基本的密钥机制。也就是说，SSH 协议要求每一个使用本协议的主机都必须至少有一个自己的主机密钥对，在服务器端通过对客户端主机密钥的认证之后，才能允许其连接请求。一个主机可以使用多个密钥，针对不同的密钥算法而拥有不同的密钥，但是至少有一种是必备的，即通过 DSS 算法产生的密钥。

每一个主机都必须拥有自己的主机密钥，密钥可以有多对，每一对主机密钥对包括公开密钥和私有密钥。在实际应用过程中怎样使用这些密钥，并依赖它们来实现安全特性呢？如图 6.14 所示，SSH 协议框架中提出了两种方案。

① 主机将自己的公用密钥分发给相关的客户机，客户机在访问主机时则使用该主机的公开密钥来加密数据，主机则使用自己的私有密钥来解密数据，从而实现主机密钥认证，确定客户机的可靠身份。从图 6.14(a)中可以看到，用户从主机 A 上发起操作，去访问主机 B



(a) 方案一

(b) 方案二

图 6.14 SSH 主机密钥管理认证方案示意图

和主机 C,此时,A 成为客户机,它必须事先配置主机 B 和主机 C 的公开密钥,在访问的时候根据主机名来查找相应的公开密钥。对于被访问主机(也就是服务器端)来说则只要保证安全地存储自己的私钥就可以了。

② 存在一个密钥认证中心,所有系统中提供服务的主机都将自己的公开密钥提交给认证中心,而任何作为客户机的主机则只要保存一份认证中心的公开密钥就可以了。在这种模式下,客户机在访问服务器主机之前,还必须向密钥认证中心请求认证,认证之后才能够正确地连接到目的主机上。

很显然,第一种方式比较容易实现,但是客户机关于密钥的维护却是个麻烦事,因为每次变更都必须在客户机上有所体现;第二种方式比较好地解决了密钥管理和维护的问题,然而这样的模式对认证中心的要求很高,在互联网络上要实现这样的集中认证,单单是权威机构的确定就是个大麻烦。但是从长远的发展来看,在企业应用和商业应用领域,采用中心认证的方案是必要的。

另外,SSH 协议框架中还允许对主机密钥的一个折中处理,那就是首次访问免认证。首次访问免认证是指,在某客户机第一次访问主机时,主机不检查主机密钥,而向该客户发放一个公开密钥的副本,这样在以后的访问中则必须使用该密钥,否则会被认为非法而拒绝其访问。

3. 字符集和数据类型

SSH 协议为了很好地支持全世界范围的扩展应用，在字符集和信息本地化方面作了灵活的处理。首先，SSH 协议规定，其内部算法标识、协议名字等必须采用 US-ASCII 字符集，因为这些信息将被协议本身直接处理，而且不会用来作为用户的显示信息。其次，SSH 协议指定了通常情况下的统一字符集为 ISO 10646 标准下的 UTF-8 格式，详细请参考 RFC-2279。另外，对于信息本地化的应用，协议规定了必须使用一个专门的域来记录语言标记(language tag)。对于大多数用来显示给用户的信息，使用什么样的字符集主要取决于用户的终端系统，也就是终端程序及其操作系统环境，因而对此 SSH 协议框架中没有作硬性规定，而由具体实现协议的程序来自由掌握。

除了在字符、编码方面的灵活操作外，SSH 协议框架中还对数据类型作了规定，提供了 7 种方便实用的种类，包括字节类型、布尔类型、无符号的 32 位整数类型、无符号的 64 位整数类型、字符串类型、多精度整数类型以及名字表类型。解释说明如下。

(1) 字节类型(byte)

一个字节(byte)代表一个任意的 8 字位值(octet)[RFC-1700]。有时候固定长度的数据就用一个字节数组来表示，写成 byte[n]的形式，其中 n 是数组中的字节数量。

(2) 布尔类型(boolean)

一个布尔值(boolean)占用一个字节的存储空间。数值 0 表示“假”(FALSE)，数值 1 表示“真”(TRUE)。所有非零的数值必须被解释成“真”，但在实际应用程序中是不能给布尔值存储 0 和 1 以外的数值的。

(3) 无符号的 32 位整数类型(unit32)

一个 32 字位的无符号整型数值，由按照降序存储的 4 个字节构成(降序即网络字节序，高位在前，低位在后)。例如，有一个数值为 63828921，它的十六进制表示为 0x03CDF3B9，在实际存储时就是 03 CD F3 B9，具体存储结构的地址分配如图 6.15 所示。

unit32首地址为：0x1000

03	CD	F3	B9
0x1000	0x1001	0x1002	0x1003

图 6.15 无符号 32 位整数类型的典型存储格式

(4) 无符号的 64 位整数类型(unit64)

一个 64 字位的无符号整型数值，由按照降序存储的 8 个字节构成，其具体存储结构与 32 位整数类似，可以比照图 6.15。

(5) 字符串类型(string)

字符串类型就是任意长度的二进制序列。字符串中可以包含任意的二进制数据，包括空字符(null)和 8 位字符。字符串的前 4 个字节是一个 unit32 数值，表示该字符串的长度(也就是随后有多少个字节)，unit32 之后的零个或者多个字节的数据就是字符串的值。字符串类型不需要用空字符来表示结束。

字符串也被用来存储文本数据。这种情况下，内部名字使用 US-ASCII 字符，可能对用户显示的文本信息则使用 ISO-10646 UTF-8 编码。一般情况字符串中不应当存储表示结束的空字符(null)。在图 6.16 中举例说明字符串"My ABC"的存储结构。

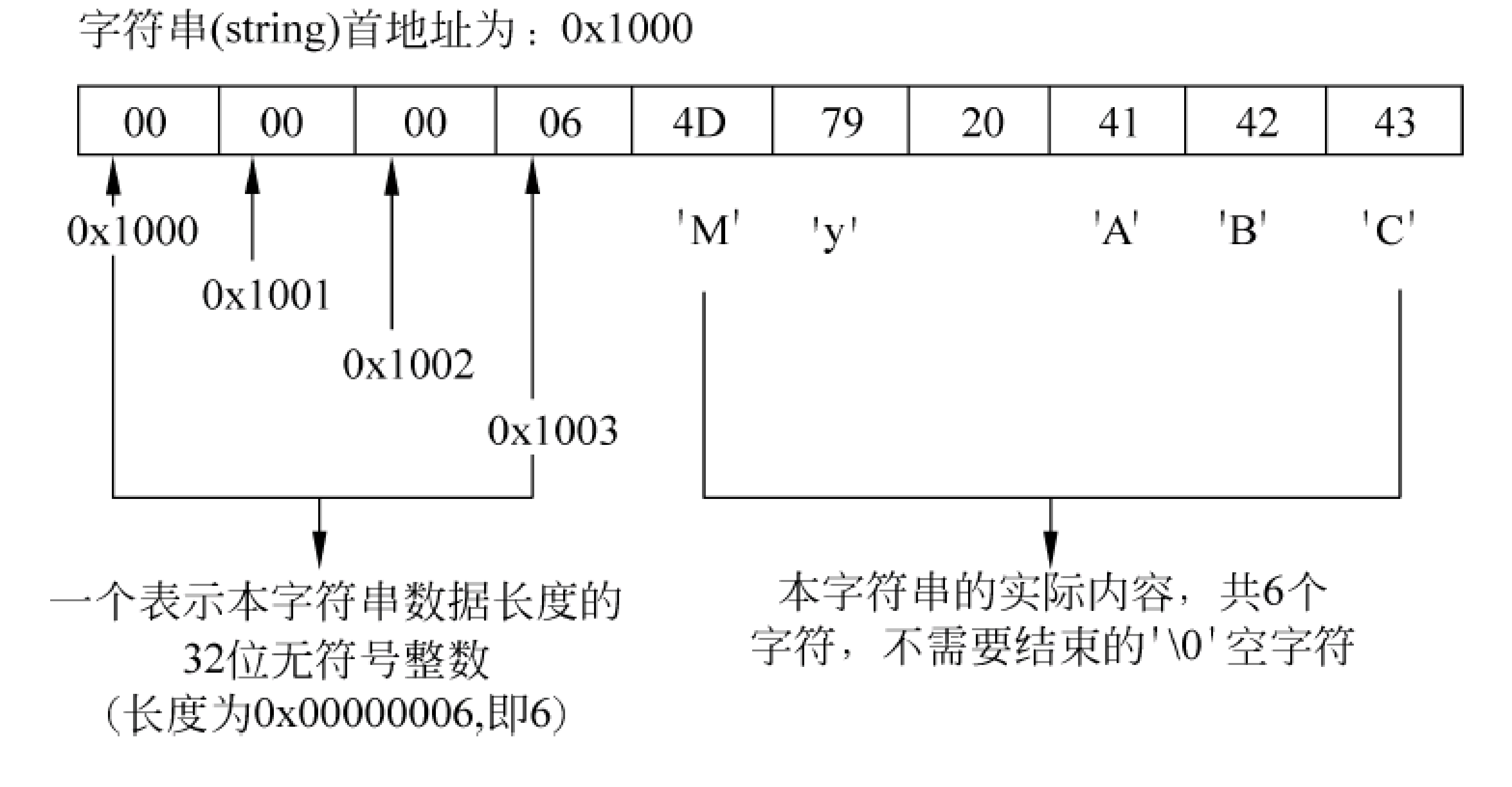


图 6.16　字符串类型的典型存储格式

从图 6.16 中可以看出，字符串类型所占用的长度为 4 个字节加上实际的字符个数(字节数)，即使没有任何字符的字符串也要占用 4 个字节。这种结构与 Pascal 语言中的字符串存储方式类似。

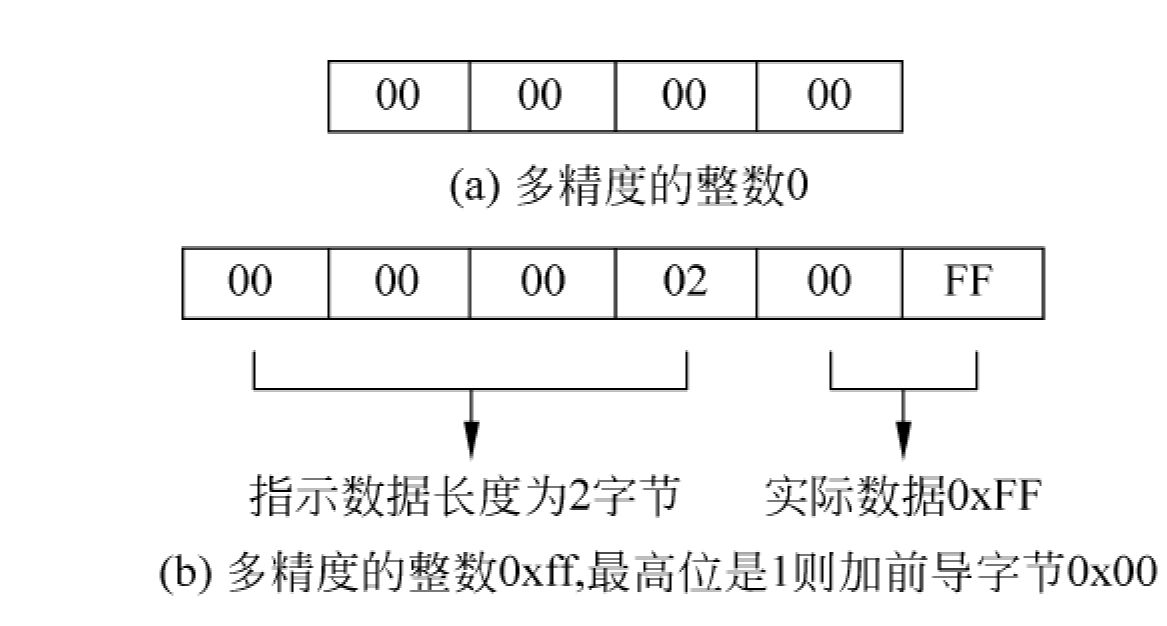
(6) 多精度整数类型

多精度的整数类型实际上是一个字符串，其数据部分采用二进制补码格式的整数，数据部分每个字节 8 位，高位在前，低位在后。如果是负数，其数据部分的第一字节最高位为 1。如果恰巧一个正数的最高位是 1 时，它的数据部分必须加一个字节 0x00 作为前导。需要注意的是，额外的前导字节如果数值为 0 或者 255 时就不能被包括在整数数值内。数值 0 则必须被存储成一个长度为零的字符串(string)。多精度整数在具体运算时还是要遵循正常的整数运算法则的。其存储格式通过图 6.17 的若干示例来说明。

(7) 名字表类型(name-list)

名字表(name-list)是一个由一系列以逗号分隔的名字组成的字符串(string)。在存储方式上与字符串一样，名字表前 4 个字节是一个 unit32 型整数，以表示其长度(随后的字节数目，类似于字符串类型)，其后跟随着由逗号分隔开的一系列名字，可以是 0 个或者多个。一个名字则必须具有非零长度，而且不能包含逗号，因为逗号是名字之间的分隔符。在使用时，上下文关系可以对名字表中的名字产生额外的限制，比如，一个名字表中的名字都必须是有效的算法标识，或者都是语言标记等。名字表中名字是否与顺序相关，也要取决于该名字表所在的上下文关系。与字符串类型一样，无论是单个的名字，还是整个名字表，都不需要使用空字符作为结束，如图 6.18 所示。

SSH 协议框架中拥有对这些数据类型的支持，这将对协议、算法的处理带来极大的便利。





(d) 多精度整数的负数表示，上面为−0x0358,补码为0xFCA8,最高位已经是1(0xFC)，不要前导空格；下面为−0xABCDEF,补码为0x543211,由于其最高不是1(0x54)，所以加前导字节0xFF

图 6.17　多精度整数类型的典型存储格式

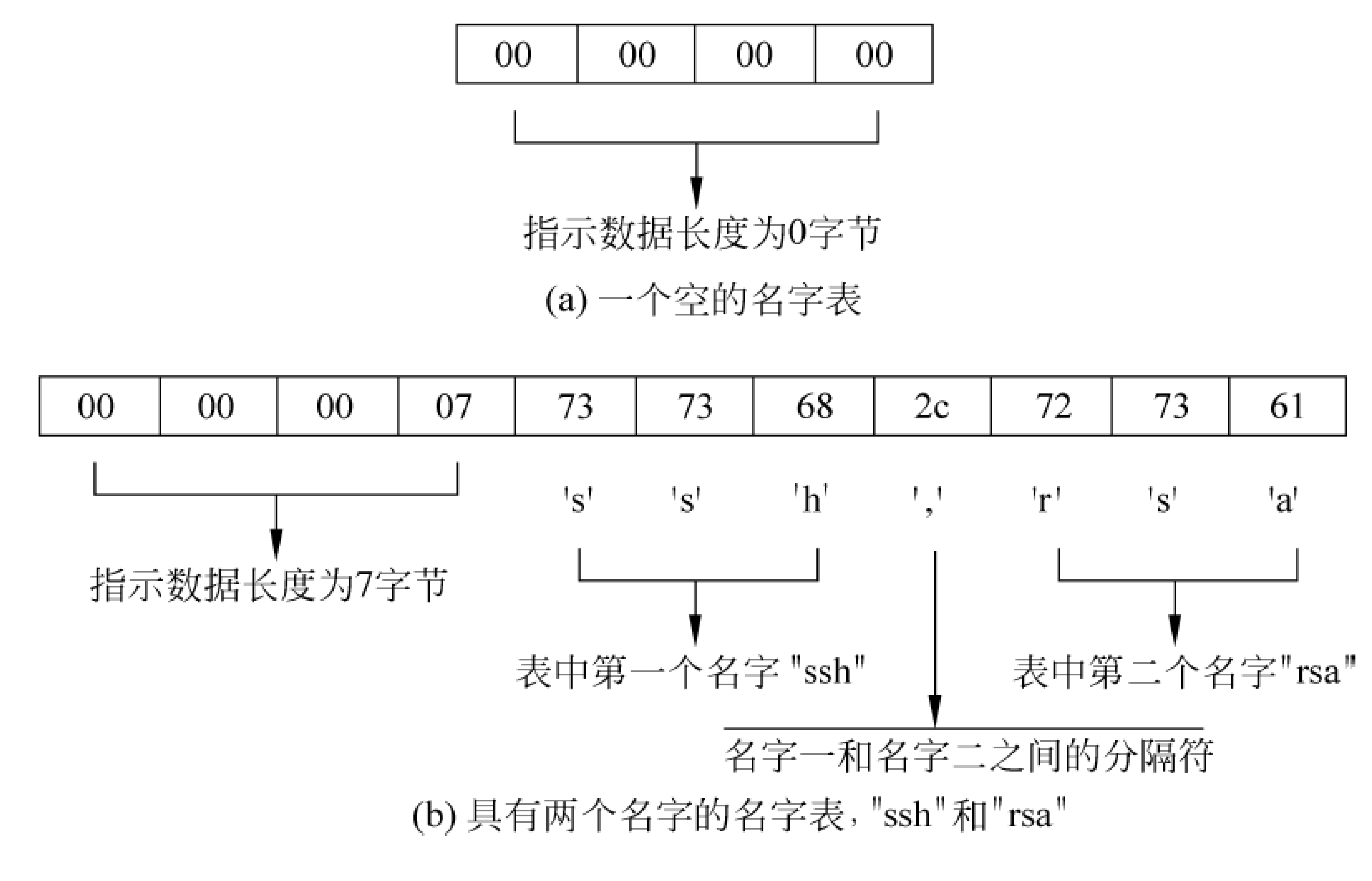


图 6.18　名字表的典型存储格式

6.2.4 SSH 协议的通信过程

在连接建立阶段，与 Telnet、FTP 等网络服务不同，SSH 要复杂得多。SSH 连接建立前要解决版本协商、会话密钥生成和认证等阶段。每个步骤具体描述如下。

（1）协议版本协商

由于 SSH 具有多种不同的版本，两个 SSH 协议首先要确认这次通信使用何种版本。具体过程是，客户端向服务端发出 TCP 请求，服务端响应客户端的 TCP 请求，并告诉客户端其自身的协议版本号和软件版本号，客户端根据服务端和自己的协议版本号和软件版本号决定使用哪种协议版本号和软件版本号进行此次通信，一般取客户端和服务端最低或互相兼容的协议版本号和软件版本号。这个过程以明文传送，还没涉及到加密。

（2）会话加密初始化阶段

SSH 通信使用会话密钥保证传输加密，这一阶段是产生会话密钥的过程。由于考虑到性能问题，会话加密采用对称加密机制。会话密钥必须安全产生，并且安全传送到另一方。SSH 协议使用公钥体系来保障会话密钥的安全传输。

具体做法是，服务端发送主机密钥公钥部分、服务密钥公钥部分、一个 64 位的随机数和支持的加密算法等信息给客户端，客户端生成会话密钥，并用服务端主机公钥、服务端服务公钥等要素对会话密钥进行加密并传送给服务端，服务端收到加密字符串后用自己的各种私钥解密，算出会话密钥。此时，会话密钥已安全传送，双方可以使用该密钥加密传输数据。

（3）认证阶段

会话密钥协商后，双方进入认证阶段。客户端首先向服务端发送用户名，服务端检查用户是否存在，如果该用户不存在则返回相应信息以示该用户不存在，如果该用户存在则通知客户“现在可以发送认证请求了”。

客户端收到“该用户存在”的信息，客户端按照已经设好的认证方式向服务端提出认证请求，对任何一个申请，如果服务端接受，服务端就发送“接受该认证”的信息给客户端，否则，以“无法识别该认证方式”回应。

SSH 可以单独使用以上的某种认证方式，也可以多种认证方式混合使用。

（4）会话模式阶段

客户端通过服务端认证后，发送会话请求，这些请求包括数据压缩、端口转发、运行 shell 和执行命令等。服务端一一审查这些请求，并返回相应信息。

认证是 SSH 最重要和最关键的阶段，下面对 SSH 认证过程进行详细分析。

SSH 协议中用户认证协议的目的是进行客户端用户的认证，它假定传输层协议是安全的，即已经验证了服务器的身份，并建立了加密的通信信道，同时也为当前会话产生了一个唯一的会话标识。在此前提下，所有认证方法中的数据包（如认证请求和应答消息）都不考虑内容保密的问题。

SSH 协议中,用户认证由服务器方主导进行。如前所述,SSH 协议支持多种认证方法。这些认证方法可以在 SSH 的认证请求消息中指明,也可以由服务器向客户机说明有哪些认证方法可以使用,之后的认证过程中,客户机可以根据情况选择一种或几种认证方法来继续认证过程。

在 SSH 协议体系中,认证方法通过名字来标识。有一个特殊的认证方法名为 none,是系统保留的。如果客户机试图通过 none 方法来连接服务器,服务器在通常情况下会将自己所支持的所有认证方法作为应答发送给该客户机。当某个客户机被免除任何认证时,服务器收到它的一个 none 请求时会无条件接受该客户机的连接请求。

无论采用什么认证方法,还是多种认证方法相结合,服务器在若干次重复的认证失败后,应该进入一个“休眠”期,短暂地拒绝访问,以使密钥的搜索攻击变得更加困难,从而在一定程度上提高协议的安全性。

SSH 的用户认证协议支持对认证方法的协商,增加了协议实现时的灵活性,增强了认证过程的可管理性,也为认证机制的配置和组合提供了可能。在协议规定的基本认证方法之外,还可以方便地扩充新的认证方法,从发展的眼光来看,这对于增加协议的生命力有着重要的意义。

1. SSH 认证请求和应答消息

SSH 协议的认证过程通过典型的请求-应答(request-response)消息来完成。请求消息由客户机发起,应答消息由服务器发起。

SSH 认证协议中的基本消息如下。

- 用户认证请求(SSH_MSG_USERAUTH_REQUEST)。
- 用户认证失败(SSH_MSG_USERAUTH_FAILURE)。
- 用户认证成功(SSH_MSG_USERAUTH_SUCCESS)。
- 用户认证标语(SSH_MSG_USERAUTH_BANNER)。

(1) 客户端认证请求

SSH 协议中,认证请求消息由客户端向服务器发起,消息内容如图 6.19 所示。

消息类型	用户名	服务名	认证方法	其他

图 6.19 SSH 认证协议的“认证请求”消息

其中,消息类型为 SSH_MSG_USERAUTH_REQUEST,代表 SSH 认证请求,包的其余部分(“其他”字段)根据具体认证方法而变化,例如可以包括一个使用公钥认证时的客户端的签名信息。服务名指定了用户所请求的服务,如果该服务不可用,则服务器必须断开连接。用户名则指明了用户的身份,如果服务器上没有该用户名,可以给客户端应答一些伪造的认证方法名,以混淆客户机的视听,避免泄露本机的账号信息。认证方法指名客户端请求的认证方式,如基于公钥的认证、密码认证和主机认证等。

在一次服务会话的认证过程中，用户名和服务名这两个域的内容是不会变化的，服务器一旦发现它们有所变化，则将此次会话中的认证状态还原，如果状态不能还原，则服务器断开连接以防欺诈。

客户机可以连续给服务器发送若干认证请求，如果其中某种认证方法不被服务器支持，则服务器就直接拒绝相应的请求。如果服务器正在处理某个认证请求的同时又收到同一客户的新的认证请求，那么旧的请求被丢弃，服务器以新请求重新开始认证。

（2）服务器认证请求应答

如果服务器拒绝了一个认证请求，它向服务器发出“用户认证失败”消息，消息格式如图 6.20 所示。

消息类型	可用的认证方法列表	认证是否成功

图 6.20　SSH 认证协议的“用户认证失败”消息

其中，消息类型为 SSH_MSG_USERAUTH_FAILURE，代表认证请求失败。认证是否成功标志的布尔量设置成“假”。可用的认证方法列表中给出一串以逗号分隔的认证方法的名字。这个消息告诉客户机：你的前一个认证请求被拒绝了，请使用所给出的认证方法名字继续尝试认证请求。

如果服务器接受并通过了客户机的认证请求，则向客户机发出“用户认证成功”消息。因为 SSH 协议允许服务器主导认证过程，并支持多种认证方法，因此服务器只是在整个认证过程全部完成后才会发送认证成功的消息。

例如，某次认证过程中，服务器需要进行 a，b，c 三种方法的认证，其认证请求和应答的典型交互过程描述如下。

① 客户机发送一个 none 认证请求。

② 服务器收到 none 请求后，向客户机发送“用户认证失败”消息作为应答，在消息的可用认证方法列表中给出 a，b，c 三种认证方法的名字。

③ 客户机收到应答后使用 a 方法再次进行认证请求。

④ 服务器使用 a 方法对客户机进行认证。

⑤ 服务器通过了 a 方法的认证后再次应答一个“用户认证失败”消息，并附带了两种认证方法，即 b 和 c。

⑥ 重复以上步骤的③，④和⑤，直到完成所有认证方法的认证后，服务器向客户机应答一个“用户认证成功”消息结束认证过程。

需要说明的是，在一次会话中，一旦服务器向客户机发出“用户认证成功”（即 SSH_MSG_USERAUTH_SUCCESS 消息）的应答之后，就会忽略此后本会话中所有的认证请求消息。如果服务器在认证成功之前收到客户机的一些非认证消息，则等到认证成功之后，这些消息才会被传递给相应的服务进行处理，相当于增加了一个缓冲。

(3) 认证过程

经过认证请求和应答之后，服务器执行认证过程，该过程和具体认证方法相关。认证过程完成后，服务器向客户端发出"认证成功"的应答消息来标记认证过程结束。即服务器在发送 SSH_MSG_USERAUTH_SUCCESS 消息之后，就开始提供客户机所请求的服务了。

SSH 协议支持在认证过程中显示标语，标语消息的格式如图 6.21 所示。

消息类型	消息内容	语言标记

图 6.21　SSH 认证协议的"标语"消息

其中，消息类型为 SSH_MSG_USERAUTH_BANNER。客户机在默认情况下将把这个消息的内容(符合 ISO-10646 UTF-8 编码)显示在屏幕上。在处理这些信息的显示时，需要采取一定的过滤措施来避免那些通过发送终端控制字符来进行网络攻击的行为，以保证安全。

以下描述 SSH 中各种认证方法的具体认证过程。

2. 基于公钥的认证

公开密钥认证方法中，拥有一个私有密钥就解决了认证问题。这种方法使用用户的私有密钥产生一个签名并发送给服务器，服务器验证签名的有效性。验证通过后，服务器接受这个认证请求，否则拒绝该认证请求。即使通过了公开密钥认证，服务器还可以要求进一步做其他认证。

SSH 协议中，典型的基于公钥的认证过程如图 6.22 所示。其中，第一步和第二步是可选的。

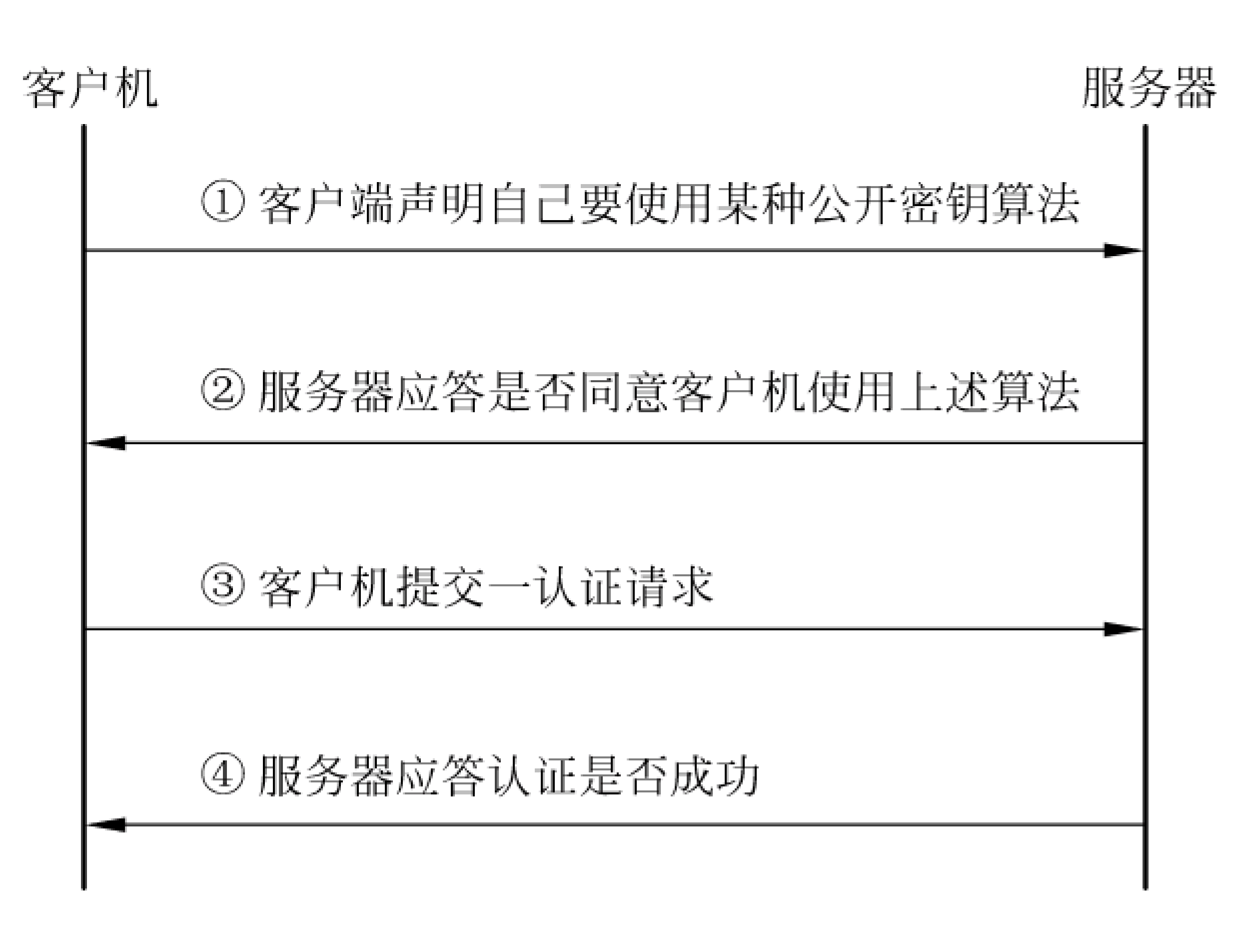


图 6.22　典型的公开密钥认证过程

① 为了避免不必要的处理和多余的用户交互，客户端向服务器发出一个"用户认证请求"消息，用来查询客户端使用的算法和密钥是否会被服务器端接受。消息内容和数据类型

如下。

```
byte      SSH_MSG_USERAUTH_REQUEST
string    用户名
string    服务名
string    "publickey"
boolean   FALSE
string    公开密钥算法名字
string    public key blob
```

公开密钥算法在 SSH 传输层协议中定义。消息中的 public key blob 中包含用户的公钥。

② 服务器收到这个消息后，按照算法名字来检查算法和公钥是否可用，如果不可用就予以拒绝，即向客户机发送一个 SSH_MSG_USERAUTH_FAILURE 应答。如果算法可用，服务器向客户端发送如下应答消息。

```
byte      SSH_MSG_USERAUTH_PK_OK
string    请求消息中指明的公开密钥算法
string    请求消息的 public key blob
```

③ 收到②中服务器的应答消息后，客户机向服务器发送一个用私钥产生的签名。客户机也可以直接发送签名而不事先去验证密钥是否能够被服务器接受，即省去以上的步骤 1 和步骤 2，直接执行步骤 3 和步骤 4。发送签名的消息其包格式如下。

```
byte      SSH_MSG_USERAUTH_REQUEST
string    用户名
string    服务名
string    "publickey"
boolean   TRUE
string    公开密钥算法名字
string    用来进行认证的公开密钥
string    签名信息
```

签名信息的内容就是用相应的私有密钥加密过的数据，数据格式如下。

```
string    会话标识
byte      SSH_MSG_USERAUTH_REQUEST
string    用户名
string    服务名
string    "publickey"
boolean   TRUE
string    公开密钥算法名字
string    用来进行认证的公开密钥
```

④ 服务器收到③中的消息后，首先检查所提供的密钥是否可以用来进行认证，如果是，验证客户端的签名。根据验证结果向客户端发送验证应答消息，即通过验证后发送“用户认

证成功”消息,否则发送“用户认证失败”消息。

3. 密码认证

SSH 协议的任何实现版本中都必须支持密码认证方法。在密码认证方法中,服务器可以根据需要要求用户修改自己的密码。典型的密码认证中,客户端向服务器发送如下“用户认证请求”消息。

```
byte    SSH_MSG_USERAUTH_REQUEST
string  用户名
string  服务名
string  "password"
boolean FALSE
string  纯文本格式的密码(ISO-10646 UTF-8 编码)
```

上面是一个用户认证的消息,它指明了认证方法是 password,即密码方法。唯一的布尔域指明是否修改用户密码,布尔值为假时,消息中只包含用户原来的密码。在认证消息中,密码采用 ISO-10646 UTF-8 编码,客户端和服务器端均可以根据需要将编码格式转换到各自环境下所需要的格式。

此外,在产生上述认证请求消息的时候,密码数据是不做加密处理的,因为 SSH 协议中的保密机制是由传输层来承担的。

通常情况,服务器对上述消息向客户端发送一个“用户认证成功”或“用户认证失败”的应答。此外,服务器也可以给出一个 SSH_MSG_USERAUTH_PASSWD_CHANGEREQ 消息,即“更改密码请求”消息作为应答。消息中包括如下字段。

```
byte    SSH_MSG_USERAUTH_PASSWD_CHANGEREQ
string  提示信息(ISO-10646 UTF-8 编码)
string  语言标记(定义于 RFC1766)
```

这个应答消息要求用户修改自己的密码。此时,客户端软件应当要求用户输入一个新的密码,然后再向服务器发送一个新的请求消息,格式如下:

```
byte    SSH_MSG_USERAUTH_REQUEST
string  用户名
string  服务名
string  "password"
boolean TRUE
string  纯文本格式的旧密码(ISO-10646 UTF-8 编码)
string  纯文本格式的新密码(ISO-10646 UTF-8 编码)
```

客户机也可以在认证一开始就发送这个消息,也就是说无论服务器是否要求用户修改密码,客户端都要求用户修改自己的密码。

密码认证方法的过程如图 6.23 所示,其中③和④是可选的。

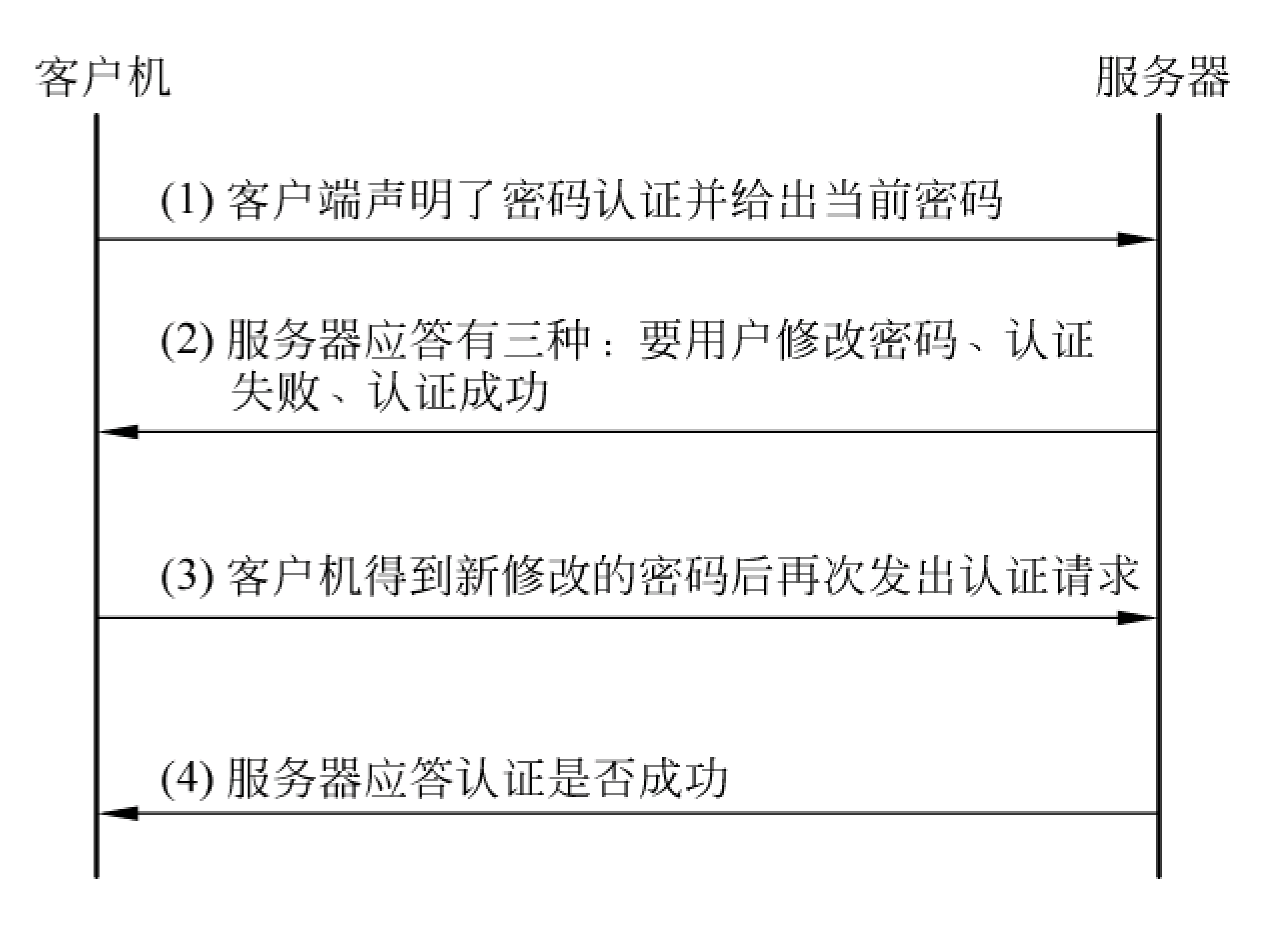


图 6.23 典型的密码认证过程

4. 基于主机的认证

基于主机的认证方法是 SSH 协议中一个可选的方法，允许在有些实现版本中不支持这种认证方法。基于主机的认证方法根据用户所在的主机和远程主机的用户名来进行认证，虽然这种认证方法在安全要求很高的场合并不适用，但它使用方便，因此应用范围广泛。如果协议支持这种认证方法，则必须保证一个普通的用户无法得到私有的主机密钥。

基于主机的认证方法中，客户机使用自己的私有主机密钥产生一个签名并发送至服务器，服务器利用该主机的公钥对其签名进行验证。客户机通过验证后，认证过程就根据服务器和客户机的用户名和主机名来进行。基于主机的认证请求消息格式如下。

```
byte      SSH_MSG_USERAUTH_REQUEST
string    用户名
string    服务名
string    "hostbased"
string    主机密钥的公开密钥算法
string    客户主机的公开主机密钥和证书
string    客户端主机名字(FQDN; US-ASCII)
string    远程客户机上的用户名(ISO-10646 UTF-8)
string    签名信息
```

请求消息中的签名信息是用私有主机密钥加密过的数据，其内容如下。

```
string    会话标识
byte      SSH_MSG_USERAUTH_REQUEST
string    用户名
string    服务名
string    "hostbased"
string    主机密钥的公开密钥算法
```

```
string  客户主机的公开主机密钥和证书
string  客户端主机名字(FQDN; US-ASCII)
string  远程客户机上的用户名(ISO-10646 UTF-8)
```

利用认证请求消息中的信息,服务器在确定主机密钥的合法性之后就可以验证请求服务的用户在主机上的身份、验证签名数据的有效性。在这种认证方法中,服务器可以选择忽略用户名而只进行客户端主机的认证。

6.2.5 SSH 协议的应用

SSH 最常见的应用就是用它来取代传统的 Telnet、FTP 等网络应用程序,通过 SSH 登录到远方机器执行各种命令。在不安全的网路通信环境中,它提供了验证机制与非常安全的通信环境。SSH 开发者的原意是设计它来取代原 UNIX 系统上的 rcp、rlogin 和 rsh 等指令程序的;但经过适当包装后,发现它在功能上完全可以取代传统的 Telnet、FTP 等应用程序。

传统 BSD 风格的 r 系列指令(如 rcp、rsh 和 rlogin)往往都被视为不安全的,很容易就被各种网络攻击手段所破解,而用来替代 r 系列指令的 SSH,则在安全方面做了强化,不但对通信内容可以进行安全的加密保护,同时也强化了对身份验证的安全机制,它应用了在密码学中已发展出来的数种安全加密机制来加强对于身份验证与通信内容的安全保护。对于消息的加密有 IDEA、three-key triple DES、DES、RC4-128、TSS 和 Blowfish 等多种安全加密算法可供选择,加密的密钥可以通过 RSA 进行交换。消息的加密可以对抗 IP spoofing,RSA 这种非对称性的加密机制则可用来对抗 DNS spoofing 与 IP routing spoofing,同时 RSA 也可以进行对主机身份的验证。

其次,通过使用 SSH 可以在本地主机和远程服务器之间设置"加密通道",并且这样设置的"加密通道"可以跟常见的 Pop 应用程序、X 应用程序和 Linuxconf 应用程序相结合,提供安全保障。

2002 年 3 月 25 日,IETF 成立专门的 Secure Shell 工作组,该组的目标是更新和标准化现行的 SSH 协议,以使 SSH 能够提供安全远程登录、安全文件传输,以及安全的 TCP/IP 和 X11 转发等服务。

目前,有关 SSH 协议的扩展 Internet 草案包括:SSH 普通消息的交换认证;SSH 文件传输协议;SSH 协议中的 GSSAPI 认证和密钥交换;SECSH 公钥文件格式;SSH 传输层协议的 Diffie-Hellman 组交换;在 DNS 中存储 SSH 主机密钥;SSH 代理转发;SSH 指纹格式等。

SSH 协议发布了两种版本,即版本 1 和版本 2。版本 1 是一个完全免费的软件包,包含几种专利算法(但其中有几种已经过期)且存在一些明显的安全漏洞(如允许在数据流中插入数据);而版本 2 安全性得到较大的提高,但在商业使用时则要付费。概括来说,SSH 协

议主要提供如下几种安全服务。

- 安全远程登录。用户可以用 SSH 完成 TELNET、RLOGIN 能够完成的任何事情。登录后所有的通信数据都受到加密保护。
- TCP 端口转发。利用 SSH 既可以进行本地端口的流量转发,也可以进行远程端口的流量转发,甚至可以结合 PPP 协议组建虚拟专用网。
- 安全远程执行命令。使用 SSH 协议,同样可调用 shell 程序,由于建立连接之后的所有数据都经过加密,因此在 SSH 建立连接后,远程执行命令时所有的通信都被加密。
- 安全远程文件传输。SSH 允许通过客户端程序 SCP 进行文件的远程复制。在 SSH 协议版本 2 中更提供了 SFTP 的安全文件传输服务。
- X 窗口连接转发。SSH 提供的一个重要功能就是 X 转发功能,它可以在客户端的显示屏上把服务器端 X 程序的运行结果以图形形式显示在客户端。

6.3 SOCKS 协议

6.3.1 SOCKS 协议概述

套接字安全性(socket security,SOCKS)是一种网络代理协议。该协议所描述的是一种内部主机(使用私有 IP 地址)通过 SOCKS 服务器获得完全的 Internet 访问的方法。具体说来是这样一个环境:用一台运行 SOCKS 的服务器(双宿主主机)连接内部网和 Internet,内部网主机使用的都是私有的 IP 地址,内部网主机请求访问 Internet 时,首先和 SOCKS 服务器建立一个 SOCKS 通道,然后再将请求通过这个通道发送给 SOCKS 服务器,SOCKS 服务器在收到客户请求后,向客户请求的 Internet 主机发出请求,得到响应后,SOCKS 服务器再通过原先建立的 SOCKS 通道将数据返回给客户。在建立 SOCKS 通道的过程中可能有一个用户认证的过程。SOCKS 位于传输层与应用层之间,使用 SOCKS 进行通信的连接建立过程如图 6.24 所示。

SOCKS 和通常的应用层代理服务器不同,应用层代理服务器工作在应用层,并且针对不同的网络应用提供不同的处理方法,比如 HTTP、FTP 和 SMTP 等,这样,一旦有新的网络应用出现时,应用层代理服务器就不能提供对该应用的代理,因此应用层代理服务器的可扩展性不好;SOCKS 独立于应用层协议,能用于多种不同的服务,它不必知道应用层协议的具体实现,一旦为双方建立相应连接,作为对等层的应用层直接实现相应的服务与数据传输。

SOCKS 可用于防火墙系统中,在该系统中,SOCKS 为客户机/服务器的 TCP 与 UDP 相关服务提供透明、安全的实现模式。一个遵循 SOCKS 协议的防火墙系统主要由 SOCKS 客户机和 SOCKS 服务器组成,如图 6.25 所示。

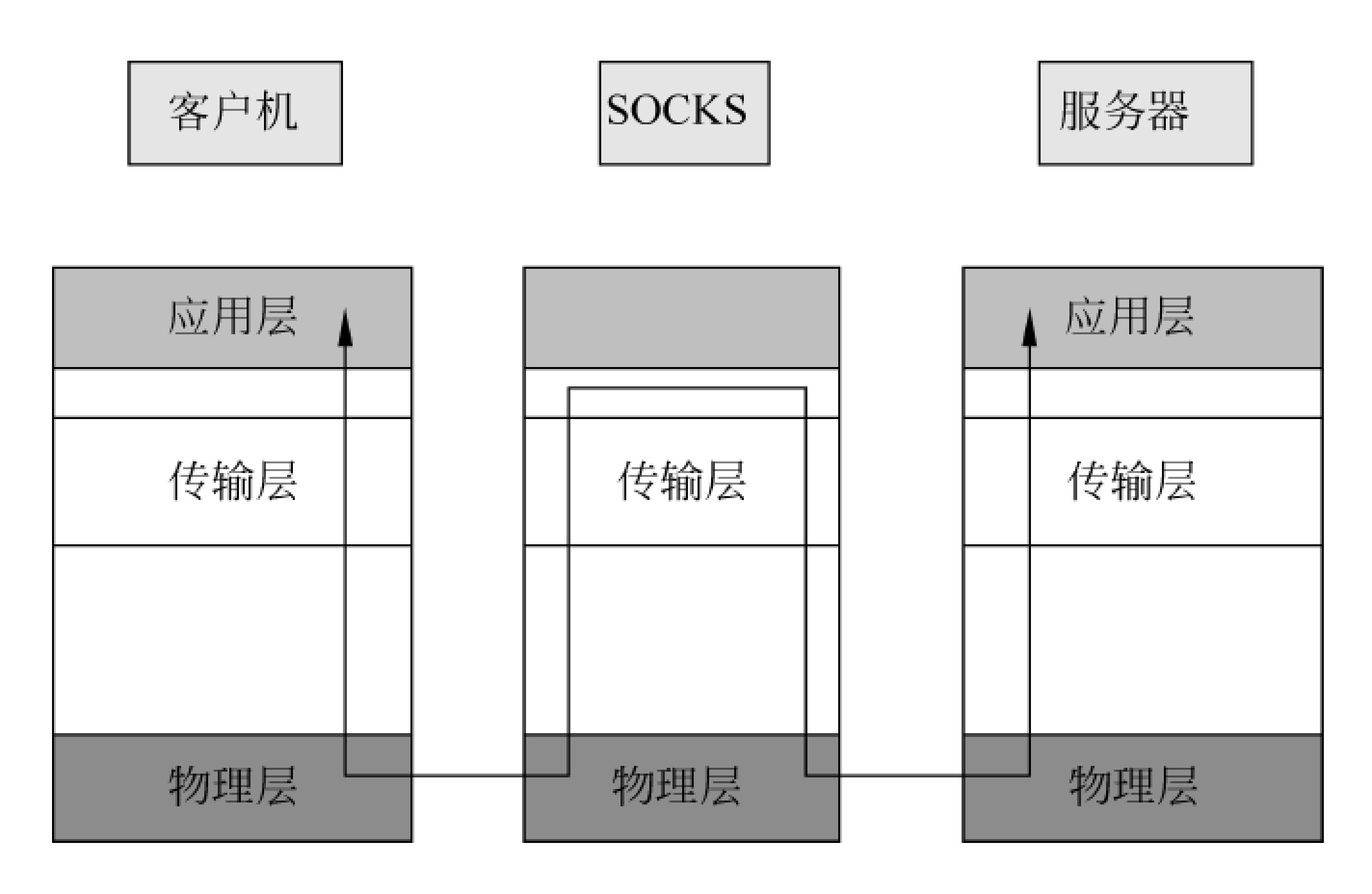


图 6.24　SOKCS 连接建立示意图

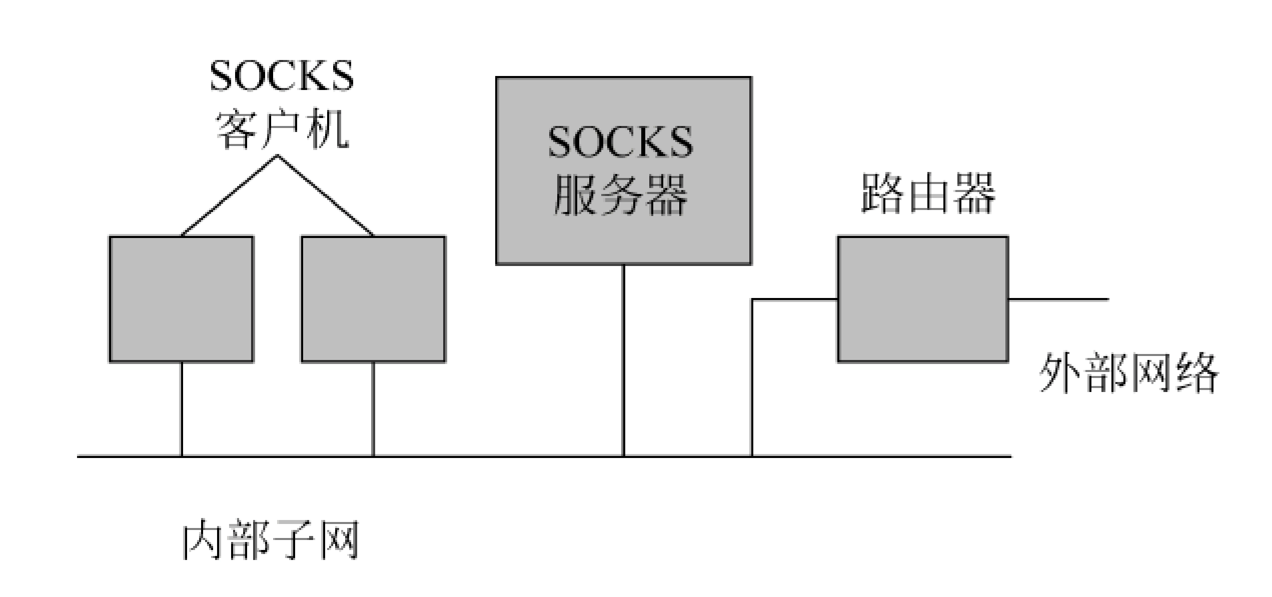


图 6.25　SOCKS 防火墙系统示意图

当前,SOCKS 协议及其应用还不断在研究与完善过程中,IETF 成立了防火墙认证传输工程组(authenticated firewall traveral,AFT)对 SOCKS 协议及相关领域进行专门研究。SOCKS 在发展过程中经历了几个版本,目前使用的是 SOCKSv5。SOCKSv5 可提供如下功能。

- 认证机制。
- 认证方法的选择与协商。
- 地址解析代理,支持 Ipv6 和 IP 地址与域名转换。
- 支持 UDP 应用程序。
- 可实现数据完整性和机密性服务。

目前,SOCKSv5 已被用于多种代理服务器产品中,如 wingate、sygate、winproxy、ccproxy 和 Microsoft proxy server 等多种代理服务器均实现并支持 SOCKSv5 协议。

6.3.2　SOCKS 协议通信过程

在一个实现 SOCKSv5 的系统中,如果客户机要同应用层服务器建立连接,首先同 SOCKS 代理服务器建立连接,应用层服务器的有关地址、端口都将在这一过程中传递给

SOCKS代理服务器，客户机与SOCKS服务器经过认证协商后，SOCKS服务器会根据SOCKS客户机请求同远程服务器建立相应的TCP或UCP连接，实现相应的应用程序协议，如图6.26所示。

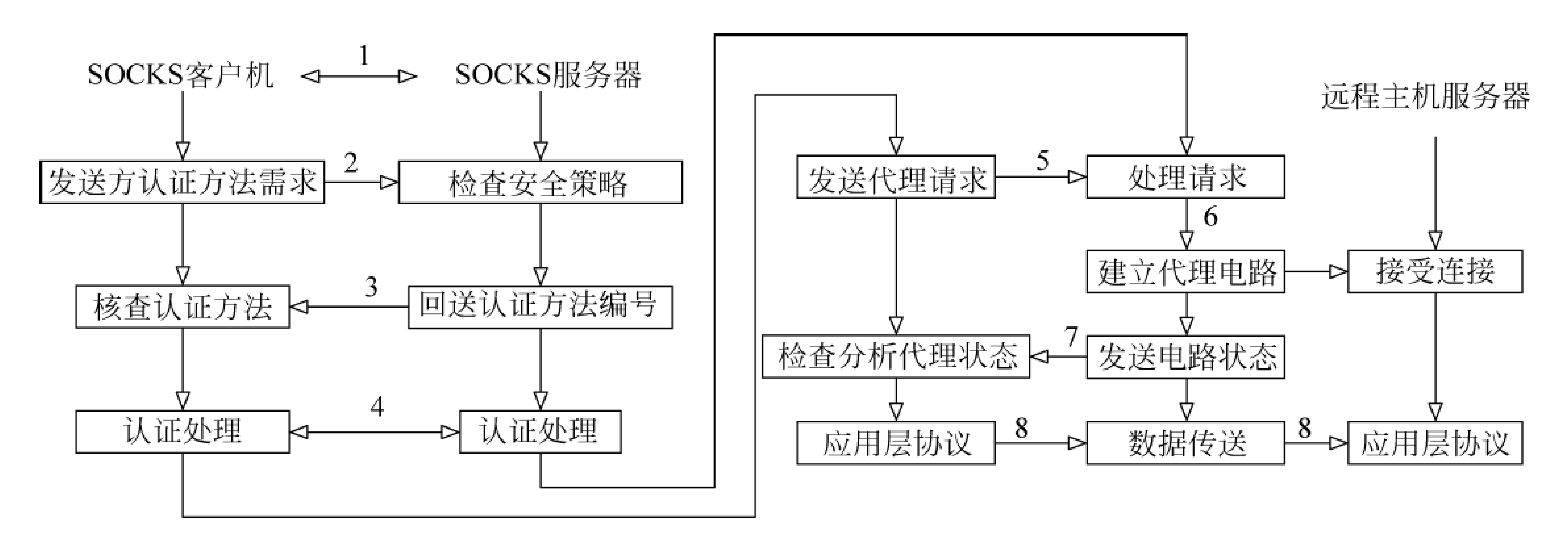


图6.26 SOCKS客户机/服务器工作原理

下面所列请求与应答格式中的十进制数代表该选项8位元长度，x'hh'代表该选项的值。SOCKS协议的通信过程描述如下。

① 首先，客户机与SOCKS服务器进行TCP连接。在通常情况下，SOCKS服务器位于TeP1080端口，如果连接成功，则进入下一步。

② 客户机向服务器发送认证请求，双方进入认证方法协商。请求消息的格式如图6.27所示。

版本号	选择认证方法号	认证方法号
1	1	1～255

图6.27 认证请求格式

其中，版本号为x'05'，表示采用SOCKSv5协议，选择认证方法号是系统支持的认证方法编号。

③ SOCKS服务器收到这个请求，经系统核实后，向客户方发送一个应答消息，消息格式如图6.28所示。

版本号	确认方法号
1	

图6.28 应答格式

其中，版本号为x'05'，方法号可以取值：x'00'表示无认证请求；x'01'表示采用通用安全服务应用程序接口；x'02'表示采用用户名/密码认证；x'03'表示采用握手认证协议(CHAP)。

④ 客户机/服务器双方在约定的认证方法后进入相应的认证处理，实现相应的认证协议。

⑤ 一旦认证成功，由 SOCKS 客户机向 SOCKS 服务器发送一个代理请求；如果认证失败，系统关闭连接，将错误写入日志。请求格式如图 6.29 所示。

版本	请求类型	保留	地址类型	目的地址	目的端口
1	1	X'00'	1	变量	2

图 6.29 代理请求格式

在随后的⑥⑦⑧步中建立代理电路，并检查分析代理状态后，最后将连接交由服务器处理。

本章实验

1. 分别在 Tomcat 和 Windows IIS 中配置 SSL，实现安全的 HTTP 服务。
2. 编程实现 HTTPS 服务器。
3. 安装支持 SOCKSv5 的代理服务器，并在其中进行安全设置。
4. 安装 SSH 的服务器和客户端程序，实现安全远程登录。

思考题

1. SSL 握手协议的作用是什么？它和 SSL 记录集协议的关系是怎样的？
2. SSL 握手过程中，假设客户机验证服务器身份，客户机通过查看服务器的证书中的时间、CA 签名、名字等判断该证书的确是由可信 CA 颁发的有效证书，此时，是否存在中间人冒充服务器进行攻击的可能？
3. SSL 协议中，密钥的计算过程是怎样的？共产生几种密钥？它们各自的作用是什么？
4. SSH 中采用哪几种用户认证方式？它们各自的特点是什么？
5. SOCKSv5 提供哪些安全服务功能？

第 7 章

应用层安全协议

7.1 Internet 的应用层安全隐患

Internet 的应用层处在 OSI 参考模型的最高层，因此任何底层协议的不安全因素都会对其安全性产生影响。应用层的目的是为网络用户提供特定的服务，如邮件传输、文件传输、WWW 服务、远程登录、域名服务和网络管理等。这些服务通过应用层协议实现，保证应用层的可用性是 Internet 的基本设计目标，然而由于 Internet 的开放性和设计上的缺陷，各种应用层协议面临严重的安全威胁。

1. 电子邮件服务

电子邮件服务通过 SMTP 协议进行传输。传统的电子邮件服务十分脆弱，通过 Internet 发送电子邮件时，电子邮件的内容是公开的，而且邮件在到达目的地之前，会经过多次转发，中途可能会经过大学、政府机构和服务提供商等各种复杂的网络环境。因为邮件服务器可接收来自任意地点的任意数据，任何人只要可以访问中间的邮件服务器，或访问电子邮件传输经过的路径，就可以截获这些信息，甚至更改、伪造电子邮件。

除了面临邮件泄密的安全威胁之外，电子邮件还可能成为黑客攻击网络的工具，如 E-mail 欺骗、E-mail 炸弹、电子邮件携带病毒等。E-mail 欺骗的潜在危害性很大：攻击者可以通过电子邮件冒充某位用户信任的权威人士(如系统管理员)，欺骗用户进行一些破坏性的或者暴露敏感信息的操作。E-mail 炸弹会在短时间内向接收者的邮箱内发送成千上万封垃圾邮件，导致邮箱的溢出，甚至令邮件服务器超负荷崩溃，产生“拒绝服务”。“特洛伊木马”和网络病毒也可携带在邮件中，对邮件接收者的系统造成危害。

2. 文件传输

FTP 是为了共享资源、方便用户进行文件下载而制定的文件传输协议，为了提供这种

服务，用户必然有对系统读写的权限，因此，攻击者常常利用 FTP 作为侵入和破坏系统的突破口。他们可能利用 FTP 将一些监控程序装入系统，以窃取管理员密码；也可能利用 FTP 获取系统的密码文件，从而了解系统的用户信息；攻击者还可以利用 FTP 的 puts 和 gets 功能，增加系统负担，严重情况下可导致系统“拒绝服务”。

匿名 FTP 是 ISP 的一项重要服务，是 Internet 上使用最广泛、信息传输量最大的应用之一。利用它可以从 Internet 上不同地点的 FTP 服务器中查询并下载各种信息资源。然而，不正确的配置将严重威胁系统的安全。FTP 的使用者可能利用这些配置上的缺陷对系统造成破坏（故意的或无意的），因此需要保证使用它的人不会申请系统上其他的区域的文件，也不能对系统做任意的修改。

对于非匿名访问，FTP 亦缺乏严格的身份鉴别机制，这些原因使得 FTP 容易成为攻击者的目标。

3. 远程登录

传统的远程登录采用 Telnet 协议实现。然而 Telnet 本身没有很好的安全保护措施，容易被攻击者利用，成为对主机及其网络进行攻击的工具。例如：Telnet 没有强力认证机制，只是简单验证连接者的用户名和密码，而远程用户的用户名和密码在网络上是明文传输的，容易被嗅探；Telnet 传输的数据没有完整性保护，面临被篡改的威胁。因此，对于入侵者而言，Telnet 可能成为一种网络攻击工具，一旦入侵者与远程主机建立了 Telnet 连接，便可以使用、更改甚至破坏目标主机上的软、硬件资源，因此，未加保护的 Telnet 服务被认为是最危险的服务之一。

4. 域名服务

DNS 采用简单的查询和应答机制进行域名解析，传统 DNS 资源记录没有保护措施，通信过程没有鉴别，因此 DNS 容易成为攻击者的攻击目标或利用它作为攻击手段对整个网络实施攻击。DNS 面临“欺骗”、“缓存中毒”、“拒绝服务”和“非授权更新”等多种安全威胁。DNS 服务器负责所在区域的域名解析，因此，对 DNS 服务器和 DNS 系统的攻击，可能导致其辖区内的网络产生严重混乱。例如，如果攻击者控制并更改了被攻击者网络系统的 DNS 服务器的 DNS 缓存，并且使用其他 IP 地址来替换其中的某个资源记录，则会将提交域名解析请求的主机引入其预先设计好的“陷阱”。

5. WWW 服务

WWW 服务易受攻击的一个原因是默认情况下该服务是完全开放的，任何人都可以在任何地方访问该服务，HTTP 协议没有身份鉴别和机密性保护机制，因此，一方面，攻击者可以利用 HTTP 开放的端口对 WWW 服务器进行攻击；另一方面，服务器上可能被放置一些恶意代码，从而对 HTTP 客户端及其主机、网络产生危害。由于浏览器一般只能理解基

本的数据格式(如 HTML、JPEG 和 GIF 等),对其他类型的数据格式,浏览器需要通过外部程序来观察,系统无法识别嵌在其基本数据单元中的有害代码,因此,这种恶意攻击难以防范。

6. 网络管理

简单网络管理协议(simple network management protocol,SNMP)主要是针对 TCP/IP 网络提出的。随着 Internet 的迅速发展,SNMP 也成为事实上的网络管理协议的标准,在互联网骨干网络设备和绝大多数厂商的网络产品中,SNMP 被广泛采用。然而,传统 SNMP 协议也存在安全漏洞,它的消息以明文形式发送,这些明文消息很容易被一些网络分析程序截取并解码,攻击者可以通过分析这些消息的内容获取有关网络资源的重要信息。SNMP 协议面临伪装、篡改、窃听和重放等多种安全威胁。例如,SNMPv1 和 SNMPv2 采用 community strings 的未加密的密码实现认证服务,攻击者可以通过网络嗅探器(如 sniffer)获取 SNMP 的社区名称,然后对网络设备及其所在网络实施攻击。

对于应用层的安全防护,可以从多个层面进行设计和实施,例如:在网络层采用 IPSec 进行端到端的保护;在传输层和应用层之间采用 SSL、SSH 以及 SOCKS 提供各种安全服务(尤其对于 Telnet 和 FTP 等可以采用 SSH 进行保护,防止密码泄露);由于应用层协议一般安装在特定的应用服务器上,对这些应用服务器的安全设置也可以在一定程度上提高应用层的安全;此外,还可以通过防火墙、VPN、入侵检测、病毒防治和漏洞扫描等综合手段共同保护应用层通信和数据的安全。

另外,还有一些专门的网络安全协议可以针对特定应用层协议进行特殊保护,例如使用 S/MIME 保护 SMTP 协议的安全;使用 DNSSEC 保护 DNS 协议的安全等;使用 SNMPv3 增强网络管理协议的安全;对于 Telnet 和 FTP 协议,目前在应用层主要采用 SSH 协议进行保护,SSH 协议见本书第 6 章。本章主要从协议级描述应用层的安全技术,重点讲述 WWW 安全技术、电子邮件安全协议、DNS 安全协议和 SNMP 安全协议等。

7.2 WWW 安全

7.2.1 WWW 安全保障体系

WWW 服务的安全是一个十分复杂的问题。由于 WWW 服务是目前 Internet 上应用最广泛的服务之一,针对它的攻击也最普遍,网络协议、操作系统以及安全管理上的任何漏洞都可能对 WWW 服务产生危害。因此,要保障 WWW 服务的安全,应该根据实际的安全需求,从不同的层面展开。

总的来说,可以把 WWW 服务的安全防护分为增强 WWW 安全机制以及保护 Web 站

点的安全两个方面进行。这两个方面的安全保障又包括许多具体的安全技术、协议和措施，如图 7.1 所示。

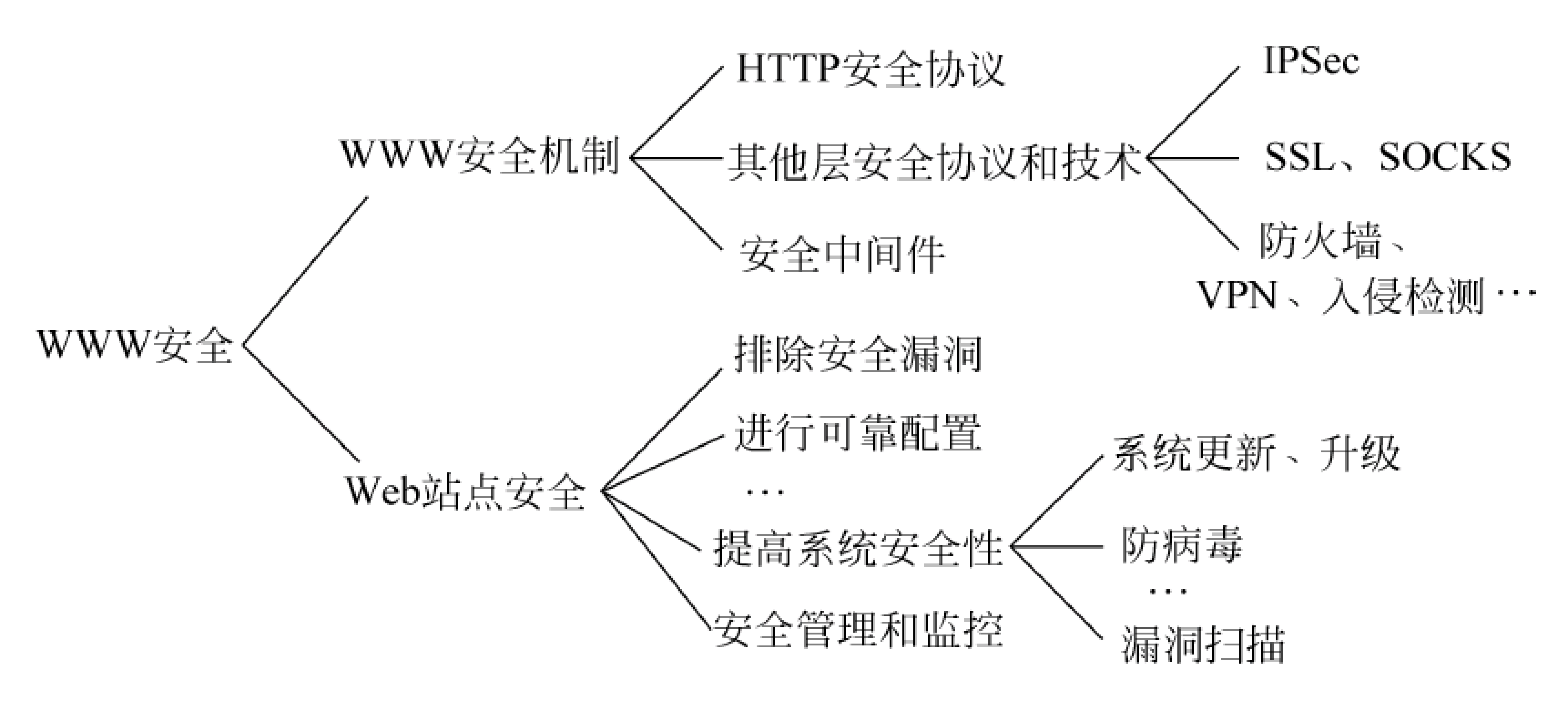


图 7.1 WWW 服务安全保障体系

图中，WWW 安全机制主要从 Internet 安全协议的层面提高 WWW 服务的安全能力，可以采用如下技术。

1. HTTP 安全协议

HTTP 安全协议包括 HTTP 协议自身的安全机制以及 SHTTP 协议两部分内容，前者涉及目前使用的 HTTP 协议提供的安全机制，主要是身份认证机制；后者通过对现有 HTTP 协议进行扩展，形成 SHTTP 协议，以提高 HTTP 协议的安全性。安全协议在 7.2.2 节讲述。

2. 其他层安全协议和技术

其他层安全协议和技术包括网络层安全协议 IPSec，传输层的 SSL 和 SOCKS，以及防火墙、VPN 和入侵检测等多种安全技术。

这些安全协议和技术是具有普适性的，即不仅可以用来保护 HTTP 协议，也可以用来保护其他应用层协议，如 FTP、Telnet 和 SMTP 等。

其中，使用传输层安全协议 SSL 保护 HTTP 协议是最常用的安全技术，SSL 协议之上的 HTTP 称为 HTTPS(HTTP over SSL)，它可以为 HTTP 应用提供基于 SSL 的认证、机密性和数据完整性等安全服务。HTTPS 需要 WWW 服务器支持 SSL 协议及公钥基础设施 PKI，这会给用户的使用带来一定的不便：例如用户很难记住自己的公钥和私钥，必须依靠某些物理设备，如 IC 卡、USK KEY 等进行密钥存储；再者，服务器和客户端必须依赖一个证书授权机构(即 CA)来签发证书，并且双方都必须将 CA 的公钥存放在本地。这对于普通 HTTP 客户端来说难以实现，因此，目前 HTTPS 协议在实际应用中往往只要求客户端鉴别服务器的证书，而不要求服务器鉴别客户端证书，在建立 SSL 安全通信后(使用 SSL 记录集协议)，再加密传输用户名和密码，以实现客户端的身份认证。

IP 层(例如采用 IPSec)和传输层的安全机制都可以为 HTTP 协议建立安全通信，但它

们都无法根据应用层传输内容的不同，提供相应的安全服务。要做到这一点，还必须依赖应用层的安全协议和技术。

3. 安全中间件

安全中间件是指由中间件提供身份认证、加密信道和访问控制等安全服务，向 HTTP 及其他应用层协议提供安全服务的应用编程接口。目前已经有不少实用的身份认证和密钥分发系统，如 MIT 的 Kerberos，IBM 的 KryptoKnight，DEC 的 SPX 等。其中 Kerberos 由于其广泛的应用已成为一个事实上的工业标准。由于应用层协议需要使用身份认证和密钥分发系统的 API，这就要求最好能有统一的 API，使得应用程序能不做修改就可以使用不同的身份认证和密钥分发系统提供的服务。这种经过标准化了的 API 称为 GSSAPI(generic security services API)。

Web 站点的安全即 WWW 服务器的安全，它为 HTTP 协议的运行提供基础环境。WWW 服务器的安全涉及安全管理、系统安全、服务器可靠配置，以及排除安全漏洞等多种安全技术和手段。

4. 排除站点安全漏洞

排除站点中的安全漏洞是最基本和有效的安全措施，应使站点中的安全漏洞降至最少。可以从以下方面减少 WWW 站点的安全漏洞。

- 防止未授权实体访问敏感数据引起的“物理漏洞”。
- 降低“错误授权”的应用程序引起的软件漏洞。例如脚本和 Applet，它们可能会执行不应该执行的功能。
- 尽量排除“不兼容问题漏洞”，它们一般由不良系统集成引起。一个硬件或软件运行时可能工作良好，一旦和其他设备集成后就可能会出现问题。这类问题很难确认，所以对每一个部件在集成进入系统之前，都必须进行测试。
- 增强密码保护和文件保护等安全策略，防止“策略漏洞”。安全策略对于增强站点的安全性至关重要，例如：如果用户使用简单密码则很容易遭受“密码猜测”攻击，因此，应该为 WWW 站点制定比较完备的安全策略。

5. 进行可靠配置

进行可靠配置指合理配置 WWW 服务器，以增强安全策略。包括：

- 合理配置服务器，使用它的访问控制和安全特性。
- 限制 WWW 用户对系统的访问权限。
- 检查驱动器和共享文件的权限，将系统设为只读状态。
- 可将敏感文件置于基本系统中，再设置二级系统，所有的敏感数据均不向互联网开放。

- 检查 HTTP 服务器使用的 Applet 脚本和客户端交互作用的 CGI 脚本，防止外部用户执行内部指令。

6. 提高系统安全性

提高系统安全性指提高站点所在的操作系统的安全性，包括及时对系统升级和安装安全补丁；安装防病毒软件和主机防火墙；进行操作系统和站点漏洞扫描等，对于发现的漏洞及时进行加固。

7. 安全管理和监控

应该对 WWW 站点访问量、故障和告警信息等进行监视和控制；同时，应该增强安全管理，例如对于新增的服务器端程序和新增的文件等进行测试和安全控制。

7.2.2 HTTP 安全协议

1. HTTP 自身的协议安全

不同版本的 HTTP 协议(包括 HTTP0.9、HTTP1.0 和 HTTP1.1)提供的安全服务能力是逐步提高的：相对于 HTTP0.9，HTTP1.0 增加了基于简单密码的基本身份认证方法；HTTP1.1 则新增了额外的报头域，对 HTTP1.0 中没有严格定义的部分作了进一步的说明。

HTTP1.0 中提供了一种基于密码的认证办法，使得 WWW 服务器可以通过“基本身份认证”支持访问控制。例如，管理员可以指定标准的 UNIX 密码文件或自己创建用户密码文件来管理用户，并形成相应的访问控制文件。当用户请求访问某个页面或运行某个 CGI 程序时，WWW 服务器读取访问控制文件，从中获得访问控制信息，并要求客户端提交用户名和密码。浏览器将用户输入的用户名和密码经过一定的编码(一般是 base64 方式)后传给服务器。在检验了用户身份和密码之后，服务器发送回所请求的页面或执行相应的 CGI 程序。用户也可以选择使用 SSL 建立加密信道后再进行身份认证，即将用户密码和密码经过编码后从 SSL 加密信道传输，这要求 WWW 服务器必须支持 SSL。

HTTP1.1 针对“基本身份认证”方法中以明文传输密码这一弱点，补充了“摘要认证方法(digest authentication scheme)”：HTTP1.1 不再传输密码明文，而是将密码经过散列函数变换以后传递其消息摘要(即密码的散列值)。使用摘要认证，攻击者不能截获密码，只能在有限的时间内进行重放攻击，这就增加了攻击的难度。为避免重放攻击，可以使用一次性的应答摘要等手段，这要求服务器记住一段时间内所有收到过的摘要值。然而，摘要认证和基本认证一样，容易受到“中间人攻击”，例如一个恶意的或被破坏的代理可能将服务器的摘要认证应答转换成基本认证应答，从而窃取用户密码。摘要认证还要求服务器存储一些用户认证信息(如用户身份等)，一旦这些信息被嗅探和窃取，攻击者可以得到这个密码保护下的所有信息。因此，摘要认证仍然不够安全。

分析可知，HTTP 协议中的“基本身份认证”和“摘要认证方法”存在潜在的安全问题：浏览器以明文的方式传递用户名和密码，或者以接近明文（编码或散列值）的方式进行密码传输，这使得 HTTP 协议仍然面临“窃听”、“假冒”等安全威胁。

2. SHTTP

安全超文本传输协议（secure hyper text transfer protocol，SHTTP）最早由 EIT 公司提出，由 RFC 2660 进行规约。SHTTP 是专门针对 HTTP 协议进行的安全扩展，可以和现有 HTTP 协议共存。它对原 HTTP 协议报头进行了扩展，形成所谓的“安全 HTTP 报头”（secure HTTP header），内含加密、鉴别和消息完整性等信息。SHTTP 使用 HTTP 的 MIME 进行签名、验证和加密，数据加密可采用对称或非对称算法。使用 SHTTP 协议的客户端在 HTTP 请求报文中，将 HTTP 头部的版本信息设置为 secure-http/1.4，支持 SHTTP 的服务器以加密和签名的消息对客户端的请求进行应答。在应答消息中，服务器可以把自己的证书及其签名信息一并发送给客户端，使客户端对服务器进行身份鉴别，服务器端可以以同样的方式鉴别客户端的身份。

SHTTP 在 HTTP1.1 的基础上提供数据机密性、身份认证、数据完整性保护和不可否认性等安全服务。SHTTP 强调的是协议的灵活性：服务器和客户端之间通过协商可以选择不同的密钥管理方法、安全策略以及加密算法等；SHTTP 支持数种消息格式标准；它不要求客户端使用公钥证书进行身份认证，相对于 SSL 而言，降低了对公钥体系的要求。

HTTPS 和 SHTTP 都是对原有通信协议作了一定修改而加入安全机制的，但由于 SHTTP 协议的复杂性，以及 SSL 协议的广泛应用，SHTTP 协议目前尚未得到广泛支持和使用。

7.3 电子邮件安全协议

7.3.1 电子邮件及其安全性概述

如图 7.2 所示，电子邮件系统通常由两个子系统组成：用户代理（user agent，UA）和邮件传输代理（message transfer agent，MTA）。用户代理是本地程序，让用户能阅读和发送电子邮件；报文传输代理将电子邮件消息（例如 SMTP 或 POP3）从源端传输到目标端。

电子邮件是 Internet 上应用最广泛的服务之一。近年来，随着 Internet 的发展，电子邮件也以惊人的速度发展，成为一种新的交流工具。人们开始使用电子邮件开展各种工作和业务。邮件应用的不断发展直接导致了邮件自身价值的不断增加。然而电子邮件系统是一个分散的系统，每封邮件都是根据一定的路由从一个 MTA 转发到另一个 MTA，几经周折后才送到用户的邮箱中，又因为传统的邮件没有加密，所以在这个过程中，邮件可能受到信

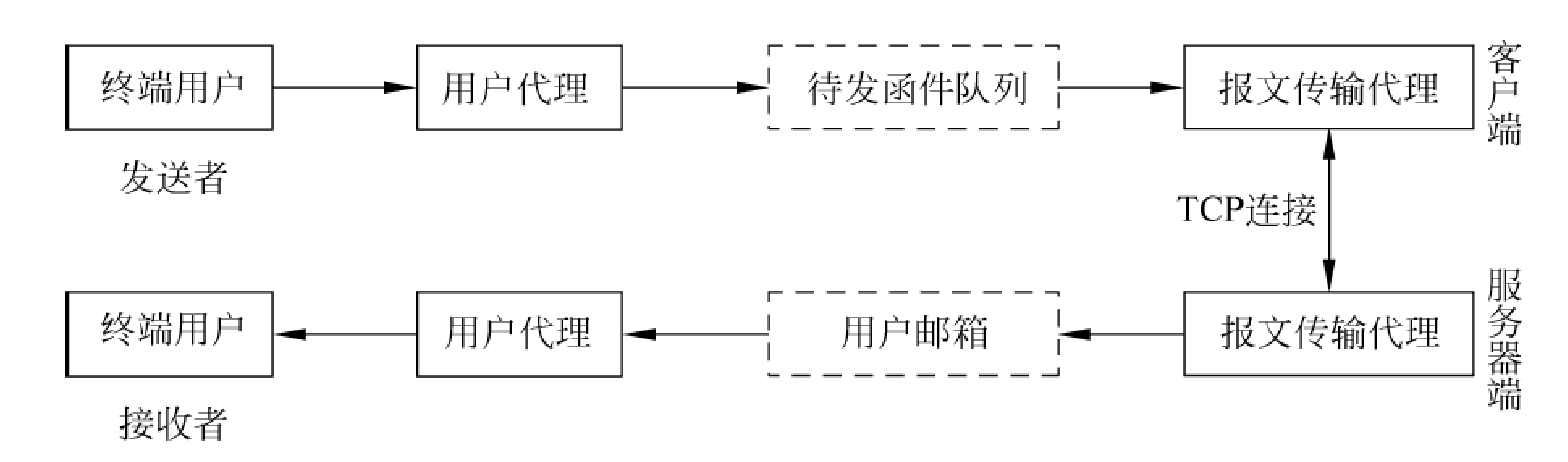


图 7.2 电子邮件在 Internet 上的传输

息泄露、内容篡改和身份假冒等威胁。

电子邮件的安全问题已经得到了人们的关注，各种各样的方案也在特定领域中发挥作用。如 PEM、MOSS、PGP 和 S/MIME 等，这些为人们提供了多种选择，但是同时也使不同方案的邮件缺乏互操作性。这些问题的产生主要是由于目前还没有确定一个实用的安全电子邮件标准造成的。

保密增强邮件(privacy enhanced mail，PEM)是由美国 RSA 实验室基于 RSA 和 DES 算法开发的安全电子邮件的早期标准，主要描述了信息格式和层次结构。PEM 只支持安全的文本信息，它的实现要求有完善的基础设施的支持，即在 PEM 的上层设施(例如认证机构 CA)没有建立起来之前，PEM 的认证框架是不具有可用性的。另外，PEM 指定了一个单一、呆板的证书层次结构，所有的 CA 都要信任同一个根 CA，但是很多组织并不想都信任同一个实体，所有这些大大限制了 PEM 的发展。

MIME 对象安全服务(MIME object security services，MOSS)针对 PEM 的不足做了一些改进，它改变了 PEM 只支持文本信息的局面，可以支持 MIME，在层次的要求方面采用了更为自由的方式。但 MOSS 有很多的执行选项，这有可能导致两个不同的开发人员提出的两个 MOSS 邮件无法沟通。可以说，MOSS 往往被认为是一种框架而不是一个规范。在实现时还要考虑许多实际的问题。

PGP 是一个软件加密程序，它既是一种规范也是一种应用，对信息的加密使用对称密码，相应的加密密钥的管理和发布则由 RSA 算法实现，同时 PGP 使用哈希算法、非对称密码实现密钥交换、信息完整性检查和数字签名。PGP 在信息加密之前进行数据压缩，可以大大减少数据的冗余度和加解密花费的时间。PGP 不是去推广一个全局的 PKI，而是让用户自己建立自己的信任网，即在 PGP 系统中，信任是双方直接的关系，或者是通过第三者、第四者的间接关系，但任意两方之间是对等的，整个信任关系构成网状结构。这样的结构既利于系统的扩展，又利于其他系统安全模式的兼容并存。但在这种信任模型中，没有建立完备的信任体系，不存在完全意义上的信任权威，缺乏有效的信任表达方式，所以它只适合小规模的用户群体，当用户数量逐渐增多时，管理将变得非常困难，用户也会发现其不易使用的一面。而且 PGP 也有其固有的缺点，从保密强度来看，PGP 的安全薄弱环节在于对加密算法(如 IDEA)的会话密钥的保护，对会话密钥采用邮件接收方的公钥加密、私钥解密。所

以,整个邮件内容的保密完全依赖于邮件接收方私钥的安全,而非发送方所能控制。另外,已经有人发现一种欺诈手段,即截获邮件后,只要对邮件重新包装并发给收件人,收件人得到的是一堆乱码,当收件人携带原件回信询问时,就可以破译加密的电子邮件。应对的这种攻击的方法是避免在回信询问时包含完整的原邮件。

S/MIME是通过在RFC 1847中定义的多部件媒体类型在MIME中打包安全服务的一种技术,可以提供验证、消息完整性、数字签名和加密。S/MIME是在PEM的基础上建立起来的,它选择了RSA实验室开发的公共密钥加密标准(public-key cryptography standard,PKCS)作为它的数据加密和签名的基础,它使用PKCS#7数据格式作为数据报文,并使用X.509v3的数字证书。S/MIME格式是建立在RFC 822中定义的双钥密码数据安全性机制之上的,其公钥管理方案是介于严格的X.509证书层次和PGP信任Web之间的混合方法。S/MIME必须对每个客户机配置可信任密钥表和证书撤销表,且证书由证书权威机构签发。在S/MIME中,认证中心具有很高的权限,能“偷窥”用户的邮件,这就要求所有的用户都必须绝对相信认证中心,同时也给电子邮件的安全带来隐患。从这一点来看,由于PGP更具保密性,所以在企业内部安全邮件的使用中,PGP的实用性更强。

PGP和S/MIME是目前电子邮件加密的两大主流技术,都是沿用IETF的标准,但两者不兼容。PGP采用了分布式的认证模式,使用比较方便,适合于公众领域和内部网络用户之间的安全信息交流;而S/MIME则采用基于CA的集中式认证模式,更适合于电子商务、政府机关和公司企业之间等对身份认证要求比较高的领域。PGP保留了用户的个人电子邮件安全服务的选择。国际电子邮件标准管理组织IMC希望形成安全邮件的统一标准。但IMC的成员意见已经并不一致。IMC只好同时发展这两种标准,直到大家有了统一的迫切要求时,再考虑统一标准之事。但从实际应用情况来看,S/MIME几乎是电子邮件厂商的首选协议,许多产品支持S/MIME,它能让用户很容易地发送和接收安全电子邮件。

由于是针对企业级用户设计的,S/MIME现在已经得到了许多机构的支持,并且被认为是商业环境下首选的安全电子邮件协议。目前市场上已经有多种支持S/MIME协议的产品。但是由于认证机制依赖于层次结构的证书认证机构,仍然不适合国内普通用户的使用。因此S/MIME协议可能作为商业和组织使用的工业标准而出现;相对来说,支持PGP的电子邮件厂商少一些。但PGP被广大的个人用户所支持和信赖,尤其是它的网状信任模型,具有很大的灵活性和适应性,大大简化了部署操作,因此对许多个人用户来说仍然具有很大的吸引力。

以下着重分析和描述S/MIME、PGP的原理和工作过程。

7.3.2 S/MIME

S/MIME(secure multipurpose internet mail extensions)协议是专门用于针对电子邮件消息进行鉴别和加密保护的应用层安全协议。S/MIME是基于RSA的MIME电子邮件格

式的安全扩展，是一种用于发送和接收安全 MIME 数据的协议。S/MIME 是从 PEM (privacy enhanced mail)和 MIME(Internet 邮件的附件标准)发展而来的是一套协议框架，它描述客户端如何创建、操作、接收和读取经过数字签名、信息加密的邮件。S/MIME 被广泛地应用于各种客户端和电子邮件平台。

针对电子邮件协议，S/MIME 提供如下安全能力。

- 对邮件内容进行加密(enveloped data)。
- 对邮件内容进行数字签名(signed data)：采用 base64 编码，这个签名的消息只有具备 S/MIME 能力的接收者才能查看。
- 对邮件进行明文签名(clear signed data)：采用 base64 编码，没有 S/MIME 能力的接收者也可看到消息内容，但不能鉴别它。
- 同时进行鉴别和加密(signed and enveloped data)。

S/MIME 采用单向散列算法(如 SHA-1、MD5 等)和公钥机制的加密体系，S/MIME 的证书格式采用 X.509 标准格式。S/MIME 认证机制依赖于层次结构的证书认证机构，所有下一级的组织和个人的证书均由上一级的组织负责认证，而最上一级的组织(根证书)之间相互认证，整个信任关系是树状结构的。

此外，S/MIME 可完成密钥生成、证书注册、证书存储和查询等密钥管理功能。

1. MIME

早期的 Internet 电子邮件有两个核心协议：由 RFC 821 定义的 SMTP(simple mail transport protocol)协议和由 RFC 822 定义的邮件格式文件。SMTP 规定了在 Internet 节点间传送或接力传送电子邮件的协议，默认使用 TCP 的 25 端口。RFC 822 定义了一种十分简单的邮件格式，这种格式的邮件只能包含纯文本信息，而且只能是 ASCII 字符，这限制了电子邮件的使用。

RFC 822 明确地把电子邮件消息分为两部分：第一部分为邮件头，其作用是标识邮件；第二部分是邮件体。邮件头中包含若干数据字段，可以在任何需要附加信息时使用。

MIME 是对 RFC 822 框架的扩充，目的是解决 SMTP 只能传输 ASCII 文本信息的局限，并约定对二进制数据进行编码的方法。MIME 协议定义了 5 个新的、可以包含在 RFC 822 报文首部的字段，分别是 MIME-Version、Content-Type、Content-Transfer-Encoding、Content-ID 和 Content-Description。

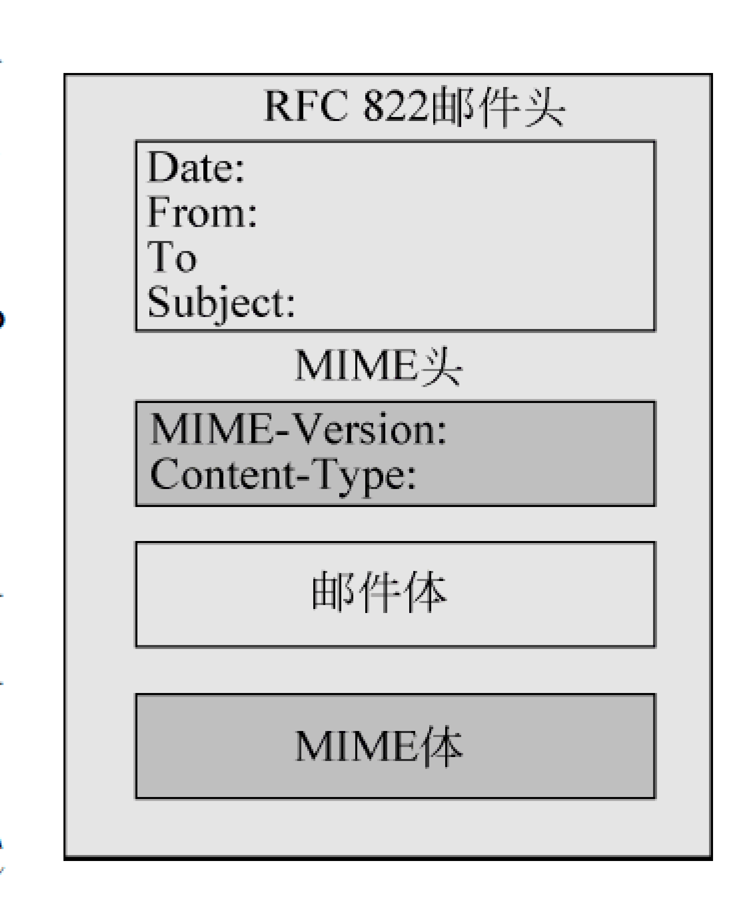


图 7.3 MIME 新增字段与 RFC 822 的结合

图 7.3 显示了 MIME 中的新增字段同标准邮件中 RFC 822 字段是如何结合在一起的。其中：

- MIME-Version 参数的值必须为 1.0，指示报文符合

RFC 2045 和 RFC 2046 的要求。

- Content-Type(内容类型)字段是必需的,该字段描述了包含在报文主体中的数据,使得接收报文的用户代理可以选择合适的代理或机制将数据向用户显示,或以一种合适的方式来处理数据。Content-Type 字段定义了 7 种基本内容类型,分别是:text、message、image、video、audio、application 和 multipart。
- Content-Transfer-Encoding 定义了两种数据编码方式,其中 base64 是一种比较常用的编码方法,它将任意二进制数据转换成一种不会被邮件传输系统破坏的格式。
- Content-ID(内容 ID)字段存在多个上下文中,是用来唯一标识 MIME 实体的标识符。

Content-Type 是 MIME 中最重要的字段,MIME 规约的大量工作集中在定义不同的内容类型上,这反映了在多媒体环境中需要提供标准化方法来处理大量不同信息类型的需求。

MIME 的 Content-Type 包括内容类型和子类型,其格式为 Content-Type:type/subtype。

内容类型说明了数据的一般类型,而子类型说明了该数据类型的特定形式。表 7.1 列出了 RFC 2046 说明的 MIME 内容类型,共包括 7 个主要的内容类型和 15 个子类型。

表 7.1 MIME 的内容类型

类　型	子类型	描　　述
Text	Plain	无格式的正文,可以是 ASCII 或 ISO 8859
	Enriched	提供了更大的格式灵活性
Multipart	Mixed	不同的部分是独立的,但是一起传输。它们应该以其出现在邮件报文里的顺序呈献给接收者
	Parrall	与 Mixed 不同的是将各部分交给接收者时未定义顺序
	Alternative	标识信息的可选择的版本。接收者的邮件系统应该将“最好的”版本显示给用户
	Digest	类似于 Mixed,但每个部分默认的类型为 Message/RFC 822
Message	RFC 822	报文主体符合 RFC 822 标准的封装格式
	Partial	以一种对接收者透明的方式对大的邮件项进行分段
	Extended-body	格式中包含了指向其他对象的指针
Image	Jpeg	图像是 JPEG 格式
	Gif	图像是 GIF 格式
Video	Mpeg	MPEG 格式的视频
Audio	Basic	单道 8 位 ISDN 编码,采样率为 8KHz
Application	PostScript	Adobe Postscript
	Octet-stream	通常的 8 位字节组成的二进制数据

2. S/MIME 对 MIME 类型的扩充

MIME 允许对基本电子邮件协议的 Content-Type 进行扩充，而 S/MIME 又在其基础上增加了几种新的 MIME 子类型，包括 multipart/signed、application/x-pkcs7-signature 和 Application/x-pkcs7-mime。

- Multipart 子类型。在 Multipart 混合类型中加入一个子类型。Signed 签名子类型标识一封经过签名的邮件，这种邮件由标准邮件部分和邮件的数字签名两部分组成。这种方法并不对邮件进行加密，因此不具备 S/MIME 功能的邮件代理也可以阅读。此时整体的内容类型字段 Content-Type 定义为 multipart/signed 类型。
- Application 子类型。S/MIME 创建了 pkcs7-mime 应用子类型来提供一些邮件安全功能，每种功能使用 pkcs7-mime 子类型中的一个单独的参数，通过 smime-type 标志来确定，smime-type 参数值有 signedData、envelopedData 等。

图 7.4 显示了加密和签名的 S/MIME 电子邮件格式，当 MIME 类型为 application/x-pkcs7-mime，smime-type＝enveloped-data 时，表示加密邮件，当 MIME 类型为 multipart/signed 时，表示签名邮件。当然，也可以对邮件进行复合，形成既加密又签名的邮件。

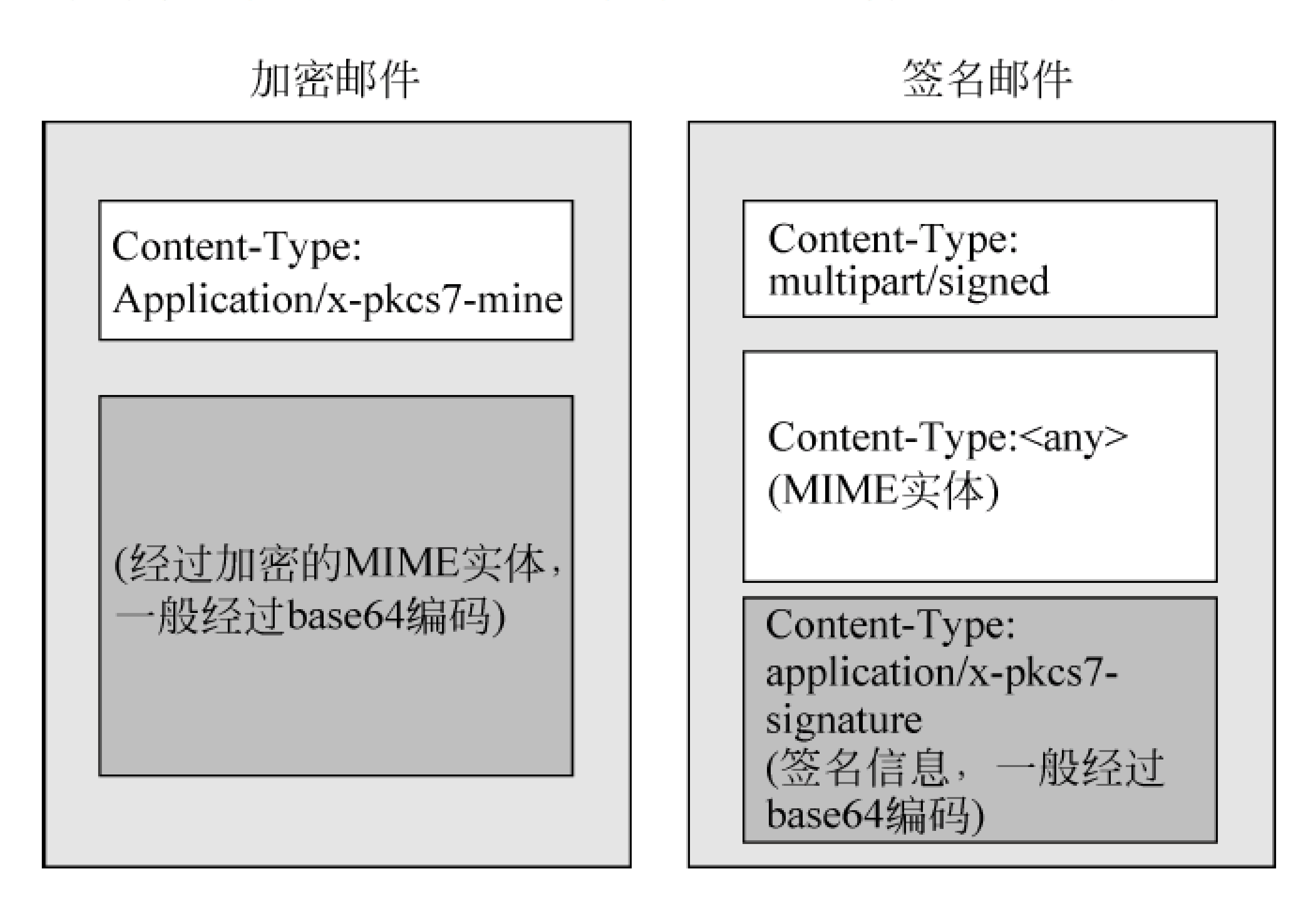


图 7.4 S/MIME 加密和签名后的邮件格式

3. S/MIME 中密码算法的应用

表 7.2 总结了 S/MIME 中使用的加密算法。

从表中可知，S/MIME 合并了如下三个公开密钥算法。

- 数字签名标准(DSS)是用于数字签名的推荐算法。
- Diffie-Hellman 是用于加密会话密钥的推荐算法。实际上 S/MIME 使用的是提供了加密/解密的 Diffie-Hellman 变体。
- 作为候选，RSA 可以既用于签名又用于会话密钥的加密。

表 7.2 S/MIME 使用的加密算法

功　　能	需　　求
创建用于形成数字签名的报文摘要算法	支持 SHA-1 和 MD5
加密报文摘要以形成数字签名	• 发送和接收代理必须支持 DSS • 发送和接收代理应该支持 RSA • 接收代理应该支持使用长度为 512～1024 位的密钥来验证 RSA 签名
加密会话密钥和报文一起传送	• 发送和接收代理必须支持 Diffle-Hellman • 发送代理应该支持使用长度为 512～1024 位的 RSA 加密 • 接收代理应该支持 RSA 解密
使用一次性会话密钥加密传输的报文	• 发送代理应支持 3DES 和 RC2/40 加密算法 • 接收代理应该支持 3DES 解密，必须支持 RC2/40 解密

对于用于数字签名的散列函数，S/MIME 建议使用 160bit SHA-1，但需要支持 128 位的 MD5。

对于报文的加密，推荐使用 3DES，但是符合标准的实现必须支持 40 位的 RC2。

S/MIME 规约包括了决定使用哪种内容加密算法的过程讨论。本质上，发送代理需要做两个决定：第一，发送代理必须决定接收代理是否能够对给定的加密算法进行解密；第二，如果接收方只能够接收弱的加密内容，发送代理必须决定使用弱加密算法是否可接收。为了支持这个决策过程，发送代理可以在它发送出去的 S/MIME 报文中按照优先选择的次序声明其解密的能力。接收代理可以存储这个信息以备将来使用。然而收发双方并不总是处在同一水平线上。例如，发送方代理可能试图使用 RC2/128 来加密 MIME 消息，而接收方可能只具有 RC2/40 解密的能力。因此 S/MIME 协议定义了一个过程，当要发送 S/MIME 消息时，该过程可以定义一个最好的算法。下面是发送方代理做决策时应该使用的一些指定的规则。

① 已知能力。如果发送代理在此前接收到了接收方一个密码(包括密码算法等)功能的列表，则发送方应该选择列出的第一个功能来加密要发送的数据。

② 未知能力但已知使用了加密。如果发送方代理对接收方代理的解密能力不清楚，但至少从接收方接收过一条曾经加了密的消息，则此时发送代理应该使用以前的那种算法来加密要发送的消息。

③ 未知能力且未知 S/MIME 版本。当发送方以前没有与接收方联系过，也不知道接收方的安全能力时，如果发送方愿意冒着接收者可能不能解密报文的危险，则应该使用 3DES 算法；如果不愿冒这个险，那么发送方使用 RC2/40。

4. S/MIME 报文处理过程

如前所述，S/MIME 使用签名、加密来保证 MIME 实体的安全。一个 MIME 实体可能是一个完整的报文(除了 RFC 822 首部)，或者 MIME 实体是报文的一个或多个子部分。MIME 实体按照 MIME 报文准备的一般规则来准备。然后，该 MIME 实体加上一些与安全有关的数据(如算法标识符和证书)后，被 S/MIME 处理以生成 pkcs 的对象。然后 pkcs 对象作为报文内容被封装成 MIME。

S/MIME 的内容类型有封装数据(envelopedData)、签名数据(signedData)、清澈签名(clearSigning)和加密且签名的数据(enveloped-and-signedData)。对不同的数据类型，其封装过程也不一样。

(1) 封装数据

准备一个封装数据的 MIME 实体的步骤描述如下。

① 为特定的对称加密算法(RC2/40 或 3DES 算法)生成伪随机会话密钥。

② 对每个接收者，使用接收者的 RSA 公开密钥对会话密钥进行加密。

③ 对每个接收者准备"接收者信息(RecipientInfo)"数据块，该块中包含发送者的公开密钥证书、用来加密会话密钥算法的标识符及加密的会话密钥。

④ 使用会话密钥加密报文的内容。

RecipientInfo 数据后面跟着加密的内容，共同组成封装数据，然后使用 radix-64 对这个信息进行编码。

为了恢复加密的报文，接收者首先去掉 base64 编码，然后使用私钥来恢复会话密钥。最后使用会话密钥解密 S/MIME 报文的内容。

(2) 签名数据

对于签名数据，准备一个 MIME 实体的过程如下。

① 选择签名算法(如 SHA 或 MD5)。

② 计算待签名内容的消息摘要。

③ 使用发送者的私钥加密报文的消息摘要。

④ 形成"签名者信息(SignerInfo)"数据块，该数据块中包含签名者的公钥证书、报文消息摘要算法标识符、用来加密消息摘要的算法标识符及加密的消息摘要等。

签名数据实体包括报文摘要算法标识符、被签名的报文和 SignerInfo。然后使用 base64 进行编码。

为了恢复签名的报文和验证签名，接收者首先要去除 base64 编码，然后使用签名者的公开密钥来解密报文摘要。接收者单独计算报文的摘要并且将它与解密后的报文摘要相比较来验证签名。

(3) 清澈签名

发送方已经签名的数据可能会被一个与 S/MIME 不兼容的接收者收到，这样会导致初

始的内容不可用。为解决这个问题，S/MIME 使用一个可供选择的结构，即 multipart/signed 类型。

Multipart/signed 类型的主体由两部分组成。第一部分可以是任意的 MIME 内容类型，以明文的形式保留并置于消息中。第二部分的内容是签名数据的一种特殊情况，称为独立签名，它省略了可能包含在签名数据中的明文的备份。

(4) 签名并加密数据

这时准备 MIME 实体的过程可以先加密数据，然后进行签名，也可以先签名再加密数据，即嵌套使用 envelopedData 和 signedData。

5. S/MIME 证书的处理

S/MIME 使用符合 X.509 版本 3 标准的公开密钥证书。S/MIME 使用的密钥管理方法是严格的 X.509 证明层次和 PGP 的信任网络的混合。S/MIME 的管理者必须为用户配置可信任的密钥表和证书废止列表，证书是经过认证机构签名的。

S/MIME 用户可以完成如下密钥管理功能。

- 密钥的生成：与管理有关的用户必须能够生成单独的 Diffle-Hellman 和 DSS 密钥对，以及 RSA 密钥对。每个密钥对必须从一个好的、不确定的随机输入源生成并且采用安全的方式进行保护。用户代理应该生成 768～1024 位之间的密钥对，并且不能生成小于 512 位的密钥对。
- 注册：用户的公开密钥和认证一起注册，获得 X.509 公开密钥证书。
- 证书的存储和查询：用户需要访问证书的本地列表来验证进入的签名和输出加密报文。

6. 增强的安全服务

可以使用三种可选的增强的安全服务来扩展当前的 S/MIMEv3 安全及证书处理服务。

- 签名收据：一种可选的服务，它考虑的是消息发送的证明。收据为发送者提供了一种向第三方出示证明的手段，接收者不仅收到了消息，而且验证了初始消息的数字签名。最后，接收者对整个消息及相应的签名进行签名作为接收的证明。该服务仅仅用于签名的数据。
- 安全标签：安全标签可以通过两种方式来使用，一是描述数据的敏感级，例如可以使用一个分级的标签列表(如机密、秘密和限制等)。二是使用标签来控制授权和访问，描述哪一类接收者可以访问数据。
- 安全邮件列表：当 S/MIME 协议提供安全服务时，发送代理必须为每一个接收者创建特定接收者的数据结构。随着某一个特定消息的接收者的数目增加，这一处理可能会降低发送消息的性能。安全邮件列表代理可以接收一个单独的消息，并针对每一个接收者完成特定于接收者的加密。

7.3.3 PGP

PGP(pretty good privacy)是由 Philip Zimmermann 设计的,可以保护电子邮件的程序。PGP 使用公钥密码、对称密码和消息完整性算法等多种密码体制,可提供认证(数字签名)、机密性、压缩、电子邮件兼容性和分段等多种服务,如表 7.3 所示。

表 7.3 PGP 服务概述

功　　能	使用的算法	描　　述
数字签名	DSS/SHA 或 RSA/SHA	消息的散列值利用 SHA-1 产生,将此消息摘要和消息一起用发送方的私钥按 DSS 或 RSA 加密
消息加密	CAST、IDEA、3DES 或 RSA	将消息用发送方生成的一次性会话密钥按对称加密算法加密。使用接收方的公钥按 Diffie-Hellman 或 RSA 算法加密会话密钥,并与消息一起发送
压缩	ZIP	消息在传送或存储时可使用 ZIP 压缩
电子邮件兼容性	基数 64 转换	为了对电子邮件应用提供透明性,一个加密消息可以用 base64 转换为 ASCII 串
分段		为了符合最大消息尺寸限制,PGP 执行分段和重组功能

PGP 对于消息的加密,分对称加密和非对称加密两种,其中非对称加密中最常用的算法是 RSA。对称加密可使用 CAST-128、IDEA 和 3DES 等多种算法。消息完整性保护可使用 SHA-1 作为哈希算法。签名算法可使用 DSS 或 RSA。本节其余部分在描述 PGP 非对称加密及签名时均以 RSA 算法为例。

和 S/SIME 不同,PGP 独立于 SMTP 协议。因此,PGP 不仅可以被用来保护电子邮件(包括正文和附件),也可被用来签名或(和)加密其他文件。PGP 不支持 X.509 证书对公钥的封装,而是采用自定义的、简单的公钥证书。

PGP 在加密前对邮件消息内容进行压缩处理,PGP 内核可以使用 PKZIP 算法压缩加密前的明文。一方面,对电子邮件而言,压缩后再经过 radix-64(即 MIME 的 base64 格式)编码可以比明文更短,这就节省了网络传输的时间和存储空间;另一方面,明文经过压缩后,相当于经过一次变换,对明文攻击的抵御能力更强。

由于 PGP 的安全、高效、易于实现和使用,它已经成为保护电子邮件最常用的方法。

1. PGP 密钥的产生和保存

由于 PGP 提供加密、签名、密钥及密码保护等多种安全服务,系统需要产生并保存各种不同的密钥,如 RSA 公钥、RSA 私钥、随机密码(密码的散列值)和会话密钥等。这些密钥是和 PGP 对等实体相关的一组密钥,即每个对称实体之间需要保留和它们相关的各种密钥。因此,由于系统中具有多对 PGP 对等实体,需要保存多组密钥。密钥需要一种安全、系

统的方法进行存储和组织，以便有效地使用。PGP 在每个节点提供一对数据结构，一个用来存储该节点与其所有对等实体之间的公钥/私钥对（即 RSA 私钥），称为私钥环；另一个用来存储该节点所知道的所有对等实体的公钥，称为公钥环。

图 7.5 说明了 PGP 中密钥对的生成过程。图的左边是用户提供的信息。

（1）公钥/私钥对的产生和保存

PGP 中，RSA 公钥用来加密会话密钥或进行邮件的签名验证，RSA 私钥用来对邮件签名或解密由公钥加密的会话密钥。

如图 7.5 所示，系统产生的随机数和用户指定的密钥长度作为素数生成的输入，使 PGP 得到两个大的素数。PGP 使用这两个素数生成一个公开密钥和一个与之关联的秘密密钥。

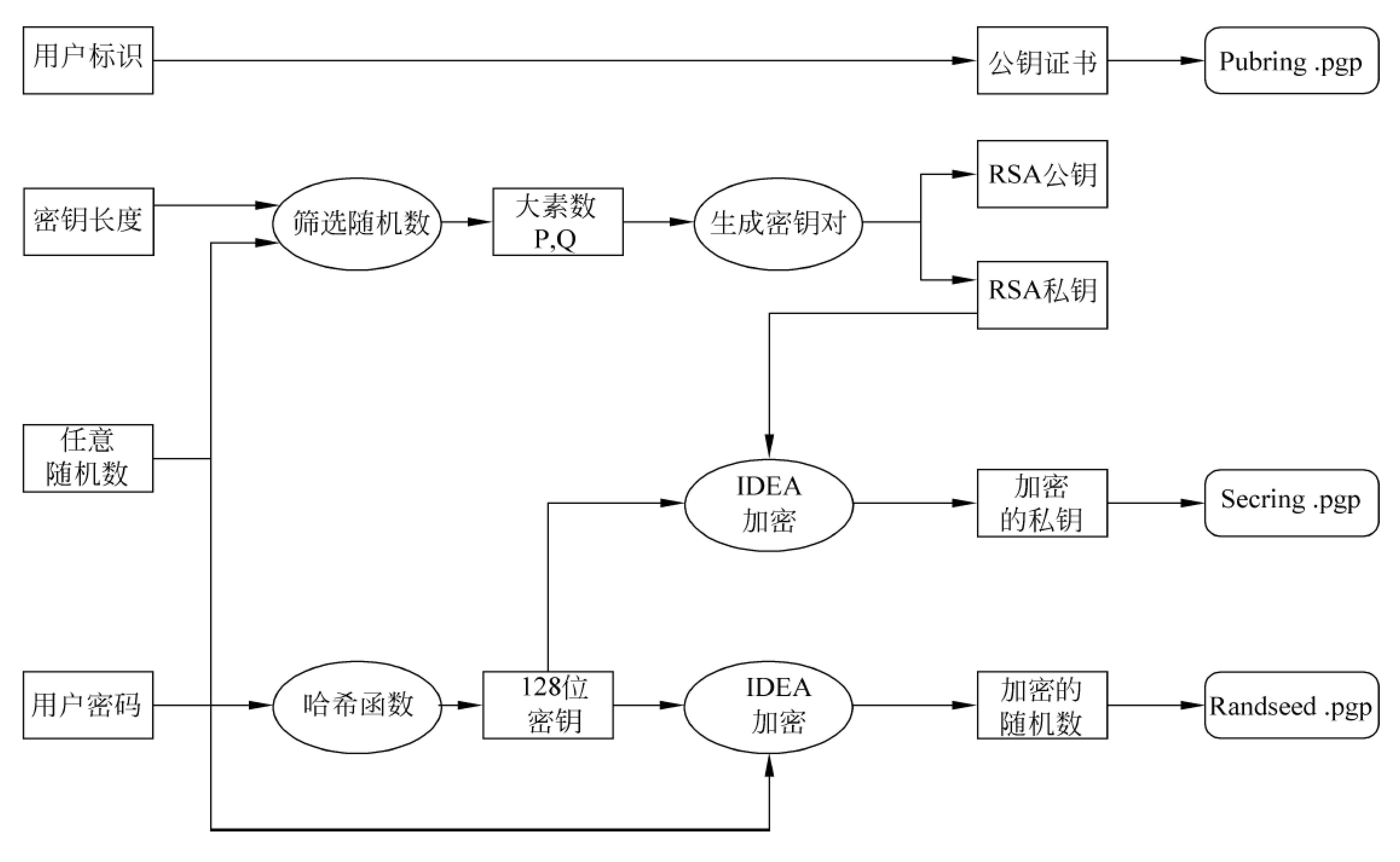


图 7.5　PGP 密钥对生成过程

上面产生的用户私钥（这里指 RSA 私钥）在 PGP 中是要加密（对称加密）保存的，加密采用对称密码，加密密钥为用户密码的散列值。一个用户可以拥有多个公钥/私钥对，以便随时更换，这些私钥构成的集合被保存在私钥环文件中。

因此，PGP 按照下面的方式保存私钥：用户密码首先经过哈希函数（如图 7.5 中的 SHA-1）产生一个 128 位的散列值。以这个 128 位的散列值作为对称密码（如图 7.5 中的 IDEA 算法）的密钥，使用对称加密算法对私钥进行加密，然后将加密后的 RSA 私钥和对应的公钥（不加密）保存到一个私钥环文件中（如图 7.5 中的 secring. pgp）。使用私钥时，需要从私钥环中取出加密的密钥，进行解密后还原出 RSA 私钥。

(2) 公钥的分发和保存

PGP 中,RSA 私钥用来对邮件签名或解密由公钥加密的会话密钥。

因此,对于用户自身的公钥,除了要把它和相应的私钥(经过密码加密)共同保存到私钥环文件中之外,PGP 还需要做如下处理:把用户标识符、公钥及其他相关信息,形成自己的公钥证书(非 X.509 格式),并将其存储到一个公钥环文件中(如图 7.5 中的 pubring.pgp 中)。

在用户配置好密钥对的时候,如果需要接收 PGP 加密信息,就必须把自己的公钥分发给对等实体。分发公钥的途径如下。

① 将自己的公钥环文件复制给别人。

② 用系统提供的功能导出公开密钥,存于文件中,然后以电子邮件或者其他方式发送给别人。

③ 使用互联网上的公钥服务器把公钥发布出去。

对于从别的用户接收到的公钥,PGP 用户将其保存到自己的公钥环文件中,以供加密或验证签名时使用。

(3) 随机数种子和会话密钥的产生与保存

PGP 中,会话密钥用来加密发送的邮件体,提供机密性服务。而会话密钥本身需要使用 RSA 公钥加密后发送给对等实体。

如图 7.5 所示,PGP 的会话密钥是一个随机数(称为随机数种子),它基于 ANSIX.917 格式,由系统中的随机数生成器产生。例如随机数生成器从用户按键盘的时间间隔取得随机数种子。

系统产生的随机数种子同样被加密(可以采用和加密 RSA 私钥相同的方式,使用用户密码的散列值进行对称加密),然后存入文件中(如图 7.5 中的 randseed.pgp)。

可以看出,PGP 每次加密都使用一个随机的会话密钥,并且进行加密保存,从而加强了 PGP 密钥系统自身的安全性,使得 PGP 可以抵抗已知明文和选择明文攻击。

2. 加密电子邮件

加密电子邮件,提供机密性服务是 PGP 的一项基本功能,需要使用非对称加密和对称加密相结合的密码体制实现。首先,密钥管理模块根据用户输入的收信人标识信息,找到收件人的公开密钥。然后,一方面,随机数发生器产生只使用一次的 128 位会话密钥,使用对称密码(如 IDEA 算法)和该会话密钥对明文邮件(一般是压缩后的)进行加密,生成密文邮件;另一方面,RSA 算法使用收件人的公钥对该会话密钥进行 RSA 加密。最后,PGP 把 RSA 加密后的会话密钥和加密后的密文邮件合并在一起,形成一个新的消息,通过 SMTP 协议发送至接收方。接收方收到邮件后首先解密出会话密钥,然后通过会话密钥解密密文邮件后还原出原始明文邮件的内容。

图 7.6 给出了发送方加密邮件和接收方解密邮件的过程。其中,发送方加密邮件的主要步骤描述如下。

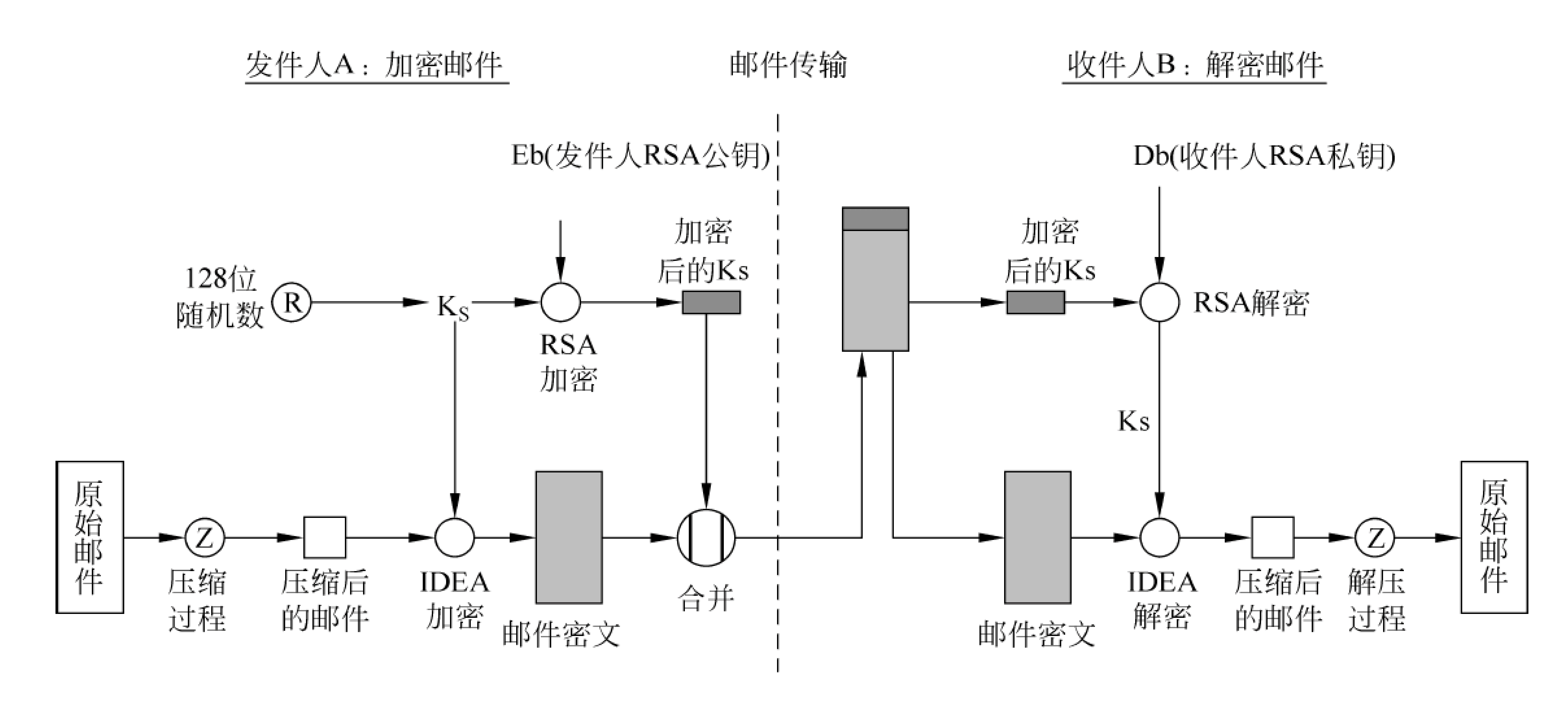


图 7.6 使用 PGP 加密和解密邮件的过程

① 生成明文电子邮件报文及用作加密该报文的随机会话密钥。

② 采用某种对称密钥算法(如图 7.6 中的 IDEA)，使用会话密钥对经过压缩后的邮件报文进行加密，形成邮件密文。

③ 采用 RSA 算法，使用接收者的 RSA 公钥对会话密钥进行加密，并附加到邮件密文前面。

收件人解密时，首先需要取得自己的 RSA 私钥：用户输入保护私钥的密码，这个密码经过散列函数(如图 7.6 中的 SHA-1)得到一个 128 位的字串。然后，PGP 把这个字串作为密钥，使用对称密钥算法(如图中的 IDEA)解密私钥环文件中加密的私钥，得到用户的 RSA 私钥。随后，接收者采用 RSA 算法，使用自己的私有密钥解密和恢复会话密钥，接着使用会话密钥解密电子邮件密文。

3. 签名电子邮件

发送方对邮件进行签名时，首先应该得到自己的 RSA 私钥，方法和上述解密过程相同，需要使用一个密码，PGP 系统将使用该密码解密私钥环文件中的 RSA 私钥。同时，用户将编辑好的邮件经过散列函数运算后得到邮件的散列值。随后，使用签名者的 RSA 私钥对其进行 RSA 加密，形成发送后的签名。最后，把邮件原文和签名合并后通过 SMTP 发送出去，从而完成签名邮件的全过程。

图 7.7 给出了发送方签名邮件和接收方验证签名的过程，图中省去了压缩邮件的过程。发送方签名邮件的主要步骤描述如下。

① 创建邮件报文。

② 从密钥环文件中取得自己的 RSA 私钥(加密的私钥)，并进行解密。

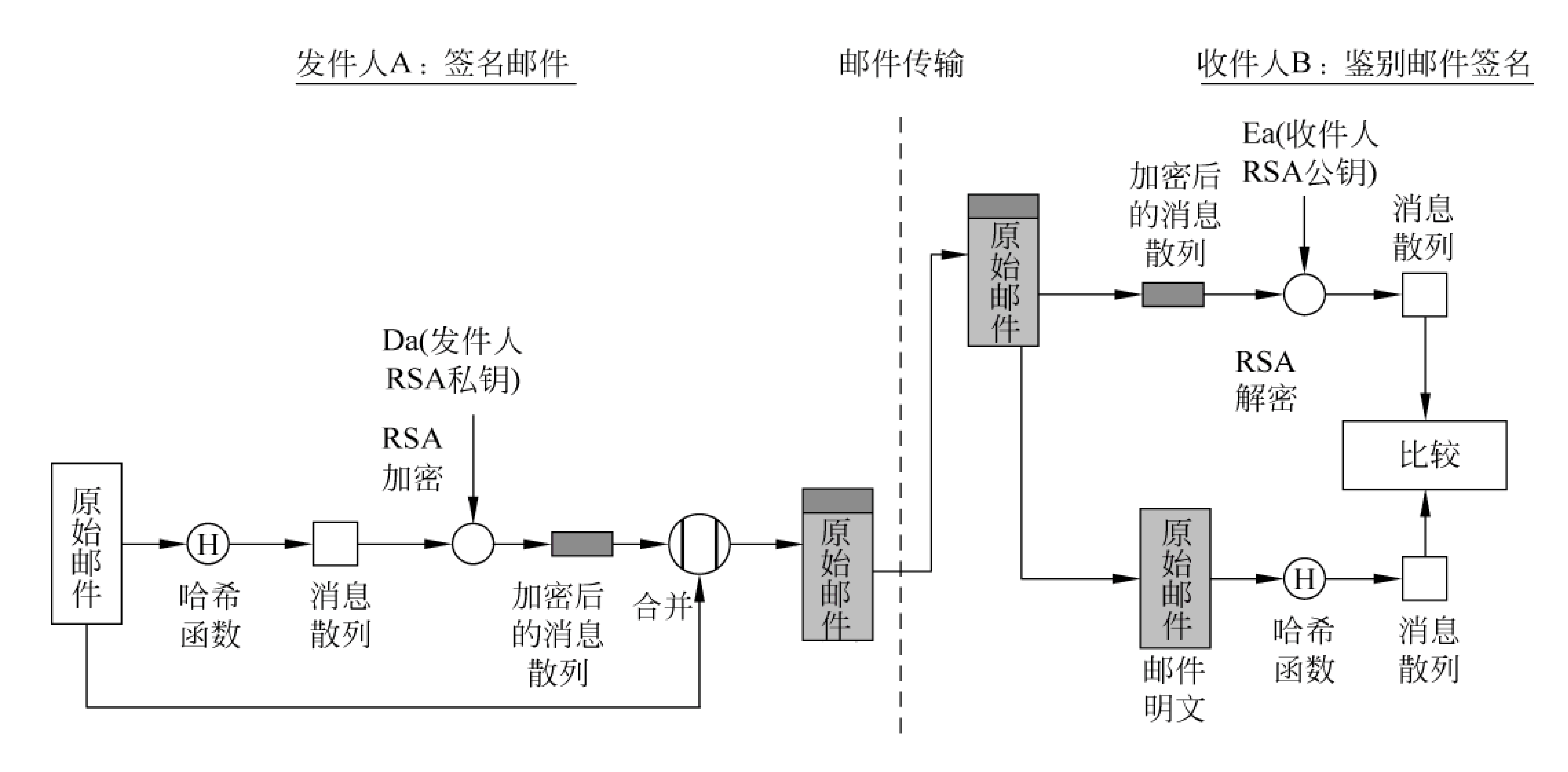


图 7.7　使用 PGP 进行签名和验证签名的过程

③ 使用散列函数(如图中的 SHA-1)生成邮件的散列值。

④ 使用自己的 RSA 私钥,采用 RSA 算法对散列值进行加密,附加在明文邮件的前面。

收件人得到带有数字签名的邮件后,需要对数字签名进行鉴别。PGP 密钥管理模块首先从公钥环文件中取出签名人的 RSA 公钥,利用 RSA 算法恢复出发信者加密的散列值。然后,PGP 重新计算明文邮件的散列值,与前者进行对比,并根据比对结果决定签名是否有效。

分析可知,PGP 加密的基本过程中使用了对称、非对称密码,而签名的基本过程中使用了非对称密码和哈希函数。当然,无论是加密还是签名,为了取得或加密 RSA 私钥,均使用对称密码及散列(哈希)函数。

可以将加密和签名一起使用,从而提供更完备的安全服务(协议的处理过程如图 7.9 和图 7.10 所示)。首先为明文生成签名并附加到邮件首部。然后使用对称密码对明文报文和签名进行加密,再使用 RSA 对会话密钥进行加密。最后,把包含签名的密文和加密后的会话密钥一起发送给收件人。

4. PGP 的消息格式

PGP 的数据信息,如加密和签名的信息、密钥证书信息、信任度信息等由一系列记录构成。每个记录包含一个数据块部分,并有一个类型标识指明此记录包含的数据块部分的含义,每个记录可以称为一个分组。如图 7.8 所示,一个记录(分组)由两部分构成:分组头部和分组体,并且分组头部和分组体的长度都是可变的。

分组头部的第一个字节称为分组类型标识(本文中,每个字节均为 8 位),它决定了分组头部的格式及分组体的内容。头部的其余部分是长度字段,它指出了分组体的长度,其长度是可变的,因此,PGP 的分组头部的长度也是可变的。分组格式的一个特例是分组长度不

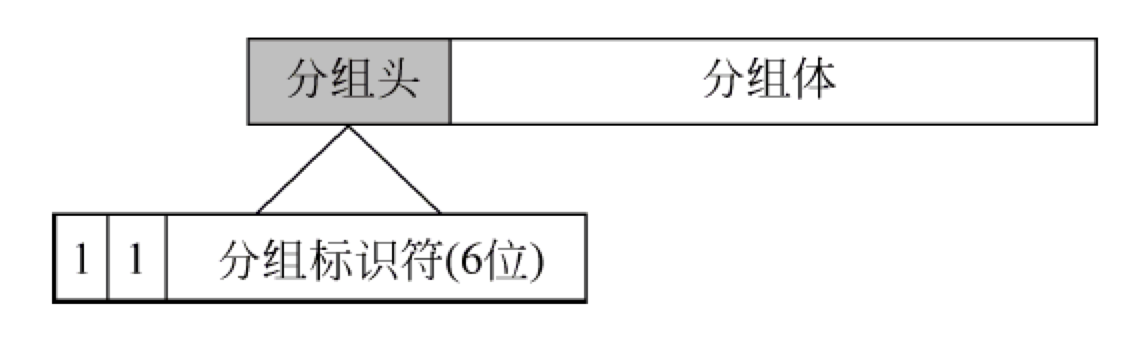


图 7.8　PGP 的分组格式

能预先知道，一个分组中将出现若干个长度字段，分组体的内容也被分割成若干个部分。

分组类型标识字段为一个字节长，如图 7.8 所示。其中最高位目前没有被使用，其值始终为 1，次高位是为了 PGP 的版本兼容性考虑。PGP 的各个版本中，存在着两个版本的信息分组格式，PGP 的当前版本为版本 4，在此版本的 PGP 程序中，次高位的值为 1。字段的其他位作为分组的标识，共可以构成 64 种类型的 PGP 分组。常用的分组包括(括号中为分组类型的编号)如下。

- 使用公开密钥加密算法加密的会话密钥分组(1)。
- 签名分组(2)。
- 私钥信息分组(5)。
- 公钥信息分组(6)。
- 压缩数据分组(8)。
- 对称加密的数据分组(9)。
- 信任度分组(12)。
- 用户标识符分组(13)。

5. 密钥标识和密钥环

如前所述，PGP 允许一个用户拥有多个公钥/私钥对，这样用户在加密或签名时可以不时改变密钥对，并且在同一时刻，多个密钥对可以在不同的通信对等实体之间进行交互，从而提高系统的安全性和灵活性。因此，PGP 中，用户和他们的密钥对之间不存在一一对应关系(这个问题在遵循 X.509 证书格式的系统中是不存在的，一般一个用户对应一个 X.509 证书，并且由证书编号来标识该证书)。这就需要某种手段来标识每次签名或加密对应的具体的密钥，以便接收者正确进行签名验证和邮件解密。例如，一个用户拥有多个公钥/私钥对时，接收者应该可以知道发送者使用了哪个公钥来加密会话密钥，以便使用相应的私钥进行解密。

为了有效标识各种密钥，PGP 系统中引入了密钥标识符(Key ID)：PGP 给每个用户的公钥/私钥对指定一个系统中唯一的 Key ID。这个 Key ID 由公钥的最低 64 位组成，这个长度足以使密钥标识符重复概率非常小。在 PGP 用户的密钥环(包括私钥环和公钥环)文件中，该密钥标识符和对应的私钥或公钥一起存储。并且，这个密钥标识符将随加密或签名后的消息一起发送给对等实体。

典型的 PGP 私钥环和公钥环的存储结构如表 7.4 和表 7.5 所示。

表 7.4 PGP 私钥环的存储结构

时间戳	密钥标识符	公钥	加密后的私钥	用户标识符

表 7.5 PGP 公钥环的存储结构

时间戳	密钥标识符	公钥	拥有者信任	用户标识符	密钥合法性	签名	签名信任

私钥环文件中有时间戳(timestamp)、密钥标识符(Key ID)、公钥和加密后的私钥、用户标识符(User ID)等字段。其中的 User ID 表示私钥的拥有者，可以是用户的电子邮件地址，也可以是一个名字。

公钥环文件中除了包括和私钥环中类似的字段外，还包括 PGP 公钥证书的内容。证书中包括签名字段，它是某个实体对该公钥进行的签名，其他字段包括拥有者信任(owner trust)、密钥合法性(key legitimacy)和签名信任(signature trust)，这些字段描述 PGP 的公钥信任度，该部分内容见"6. PGP 信任关系"中的信任模型。

图 7.9 和图 7.10 中描述了 PGP 中发送和接收经过签名的加密邮件时，公钥环、私钥环的操作，以及加密、签名的完整过程，图中的发件人为 A，收件人为 B。

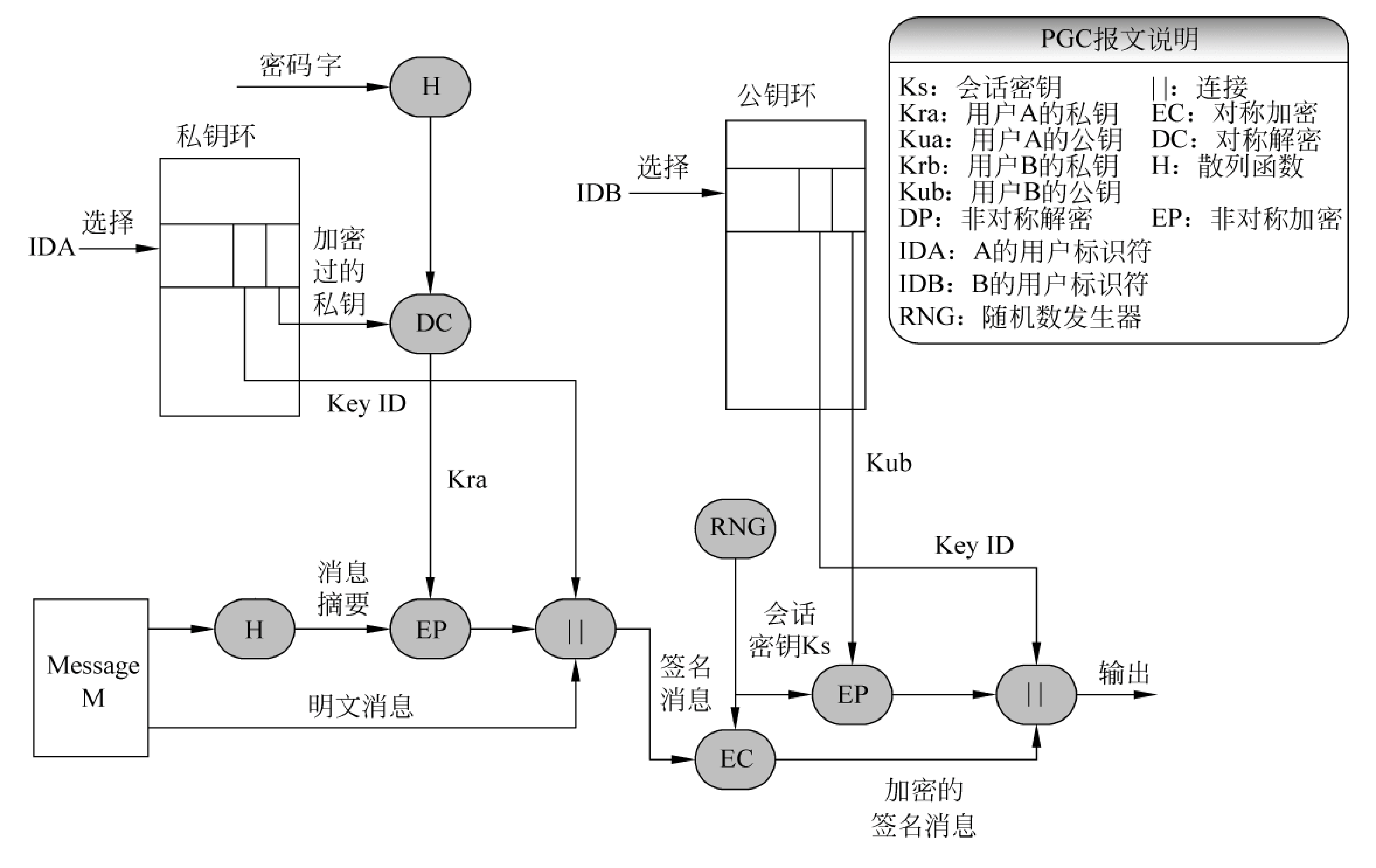


图 7.9 发送经过加密的签名邮件的过程

发送方首先为邮件产生消息签名，然后加密签名消息。用户 A 根据自己的用户标识符从其私钥环中取出经过加密的私钥，系统对加密私钥进行解密后还原出用户 A 的私钥。随后，用户 A 使用该私钥对邮件消息进行签名，同时从自己的私钥环中取出该密钥对应的 Key ID，附加在签名后的邮件消息中。发送方利用随机数生成器生成加密消息使用的会话密钥，并使用会话密钥加密签名后的邮件。然后，根据接收方的用户标识符从自己的公钥环中取出接收方的公钥，并使用该公钥加密会话密钥。同时，从自己的公钥环中取出该密钥对应的 Key ID。发送方将加密的会话密钥和加密后的签名邮件，以及接收方的 Key ID 合并，结束消息处理过程。注意，发送方的消息中含有发送方和接收方的密钥标识符。

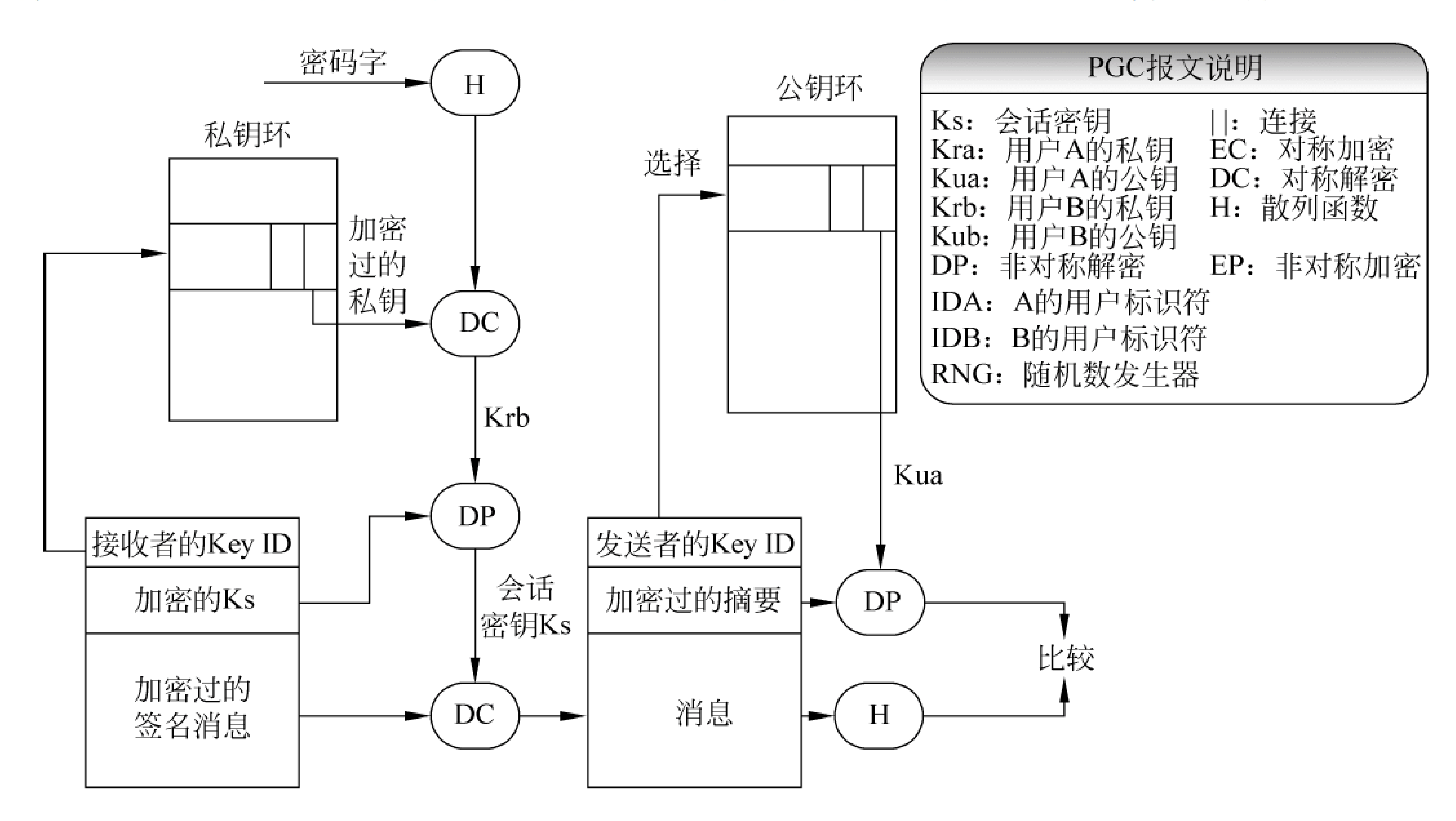


图 7.10 接收加密的签名邮件后的处理过程

在接收方，用户 B 首先解密消息，然后验证签名：用户 B 根据消息中的 Key ID（接收方的 Key ID）从自己的私钥环中取出经过加密的私钥，系统对加密存储的私钥解密后还原出用户 B 的私钥。随后，接收方利用私钥解密出会话密钥，进而使用会话密钥解密加密的签名邮件。最后，接收方根据签名消息中附加的 Key ID（发送方的 Key ID）从自己的公钥环中取出发送方的公钥，对签名进行鉴别，完成消息处理过程。

6. PGP 信任关系

（1）公钥可信度对系统安全性的影响

由于没有可信的第三方签名和颁发公钥证书，PGP 中公钥的发布可能存在安全性问题，例如公钥被篡改导致使用的公钥与公钥持有者的公钥不一致。这在公钥密码体系中是

很严重的安全问题。因此必须帮助用户确认使用的公钥是可信的。公钥信任管理可以克服PGP 中密钥分配不安全、不方便的缺点。

以用户 A 和用户 B 通信为例，现假设用户 A 想给用户 B 发送电子邮件。首先，用户 A 必须获取用户 B 的公钥，用户 A 通过下载或其他途径得到 B 的公钥，并使用它加密邮件，然后把加密后的邮件发送给 B。

此时，攻击者 C 潜入网络中，侦听并截获了用户 B 的公钥，然后在自己的 PGP 系统中以用户 B 的名字生成密钥对中的公钥，替换了用户 B 的公钥，并放在网络上或直接以用户 B 的身份把更换后的用户 B 的“公钥”发给用户 A。A 用来发送邮件的公钥是 C 伪装 B 生成的公钥(A 得到的 B 的公钥实际上是 C 的公钥/密钥对，用户名为 B)。这样一来，B 收到 A 的加密邮件后就不能用自己的私钥解密该密文邮件了，导致系统的混乱。用户 C 还可伪造用户 B 的签名给 A 或其他人发信，因为 A 手中 B 的公钥是假冒的，用户 A 会以为该邮件的确来自用户 B。于是 C 就可以用他手中的私钥来解密 A 给 B 的信，还可以用 B 真正的公钥来转发 A 给 B 的信，甚至还可以改动 A 给 B 的信。

防止篡改公钥的方法有多种，例如可以直接从对方的手中得到其公钥。此外，还可以通过电话认证密钥，如在电话上以 radix-64 的形式口述密钥或密钥指纹，密钥指纹(keys fingerprint)是 PGP 生成密钥的 160 位的 SHA-1 摘要(16 个 8 位十六进制)。这两种方法均有使用不便的局限性。此外，可以像 X. 509 证书体制那样，引入由一个用户信任的机构担当第三方，即“认证机构”，然而这样的“认证机构”适合由非个人控制的组织或政府机构充当，以注册和管理用户的密钥对。对于那些非常分散的个人用户，PGP 更赞成使用私人方式的密钥转介，因此这种第三方信任的方式在 PGP 中难以得到实际应用和推广。

(2) PGP 的信任模型

虽然 PGP 没有关于建立认证权威机构或建立信任体系的说明，但它提供了一个利用信任关系的手段，将信任与公钥关联。PGP 为公开密钥附加信任和开发信任信息提供了一种方便的方法，通过附加在公钥证书或公钥环中的各个字段来实现。

如图 7.11 所示，公钥环的每个实体都是一个公开的密钥证书。与每个实体相联系的是密钥合法性(key legitimacy)字段，用来指示 PGP 信任“这是一个合法的用户公开密钥”的程度。信任程度越高，这个用户标识符与这个密钥的绑定就越紧密。这个字段由 PGP 计算得出。

与每个实体相联系的还有用户收集的多个签名。每个签名都带有签名信任(signature trust)字段，用来指示该 PGP 用户信任签名者对这个公开密钥证明的程度。密钥合法性字段是从这个实体的一组签名信任字节中推导出来的。最后，每个实体定义了与特定的拥有者相联系的公开密钥，包括拥有者信任(owner trust)字段，用来指示这个公开密钥对其他公开密钥证书进行签名的信任程度(这个信任程度是由该用户指定的)。可以把签名信任字段看成是来自于其他实体的拥有者信任字段的副本。

图 7.11 中给出了一个正在处理的公钥环的例子，操作描述如下。

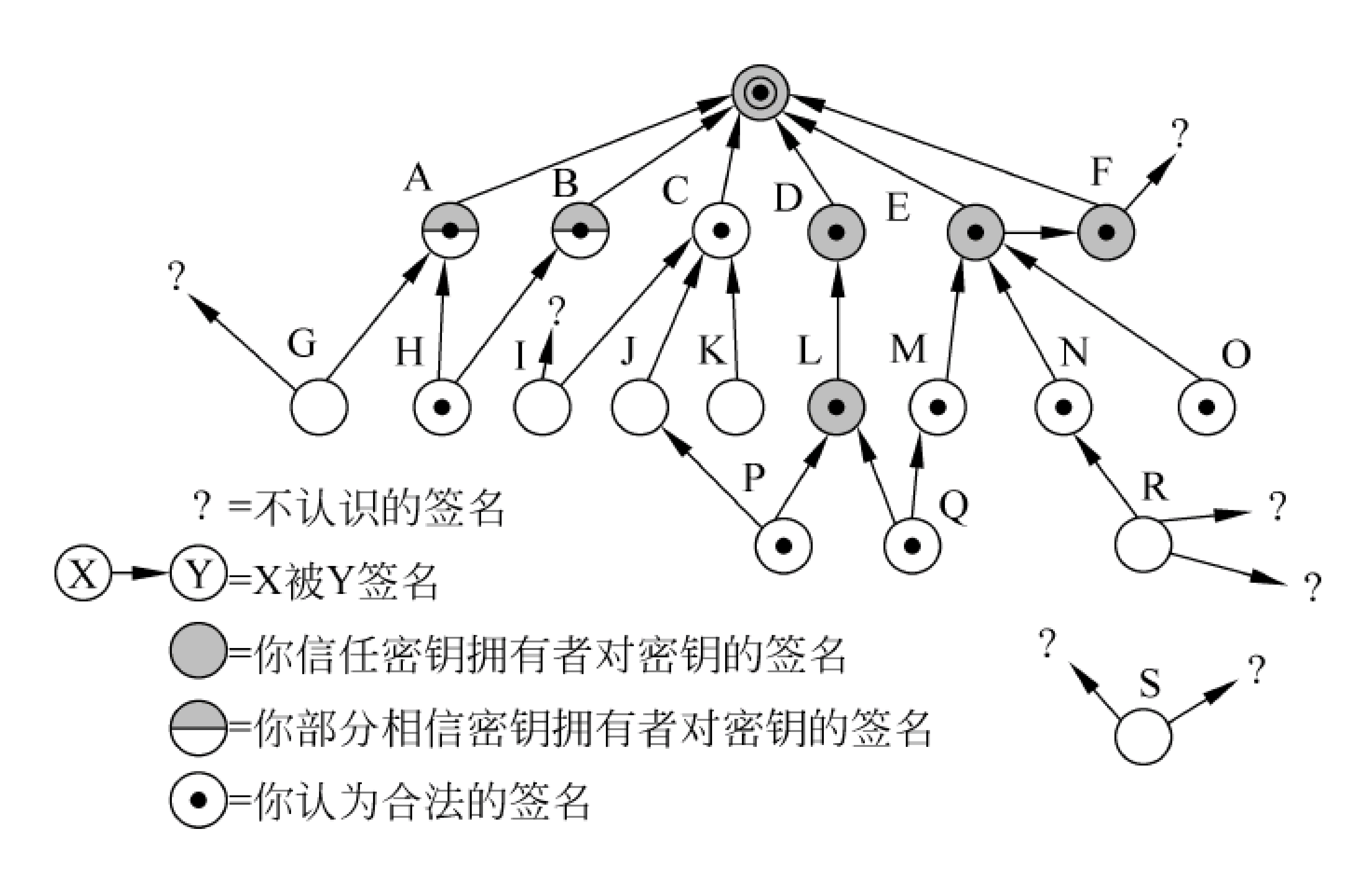


图 7.11 PGP 中的信任关系

当 A 在公开密钥环中插入了新的公开密钥时，PGP 为与这个公开密钥拥有者相关联的信任标志赋值，插用户 A 的公钥，若赋值为 1 表示终极信任；否则，须说明这个拥有者是未知的、不可任信的、少量信任的和完全可信的等，赋以相应的权重值 1/x、1/y 等。

当新的公开密钥输入后，可以在它上面附加一个或多个签名，以后还可以增加更多的签名。在实体中插入签名时，PGP 在公开密钥环中搜索，查看这个签名的作者是否属于已知的公开密钥拥有者。如果是，为这个签名的 Signature Trust 字段赋予该拥有者的 Owner Trust 值。否则，赋以不认识的用户值。

密钥合法性字段的值是在这个实体的签名信任字段的基础上计算的。如果至少一个签名具有终极信任的值，那么密钥合法性字段的设置为完全；否则，PGP 计算信任值的权重和。对于总是可信任的签名赋以 1/x 的权重，对于通常可信任的签名赋以权重 1/y，其中 x 和 y 是用户可配置的参数。当介绍者的密钥/User ID 绑定的权重达到 1 时，绑定被认为是值得信任的，密钥合法性被设置为完全。因此，在没有终极信任的情况下，需要至少 x 个签名总是可信的，或者至少 y 个签名是可信的，或者是上述两种情况的某种组合。

对以上信任模型分析可知，PGP 采用通过信任签名者的签名来间接信任用户公钥的方式建立公钥的信任度。

假设 A 和 B 有一个共同的朋友 D，而 D 知道他手中 B 的公钥是正确的。于是 D 签名 B 的公钥并将其上传到 BBS 上提供其他用户下载。A 想要获得 B 的公钥就必须先获取 D 的公钥来解密该公钥(B 的公钥)的签名，这样就等于增加了一层可信度。如果 A 信任 D，并已验证了 D 的签名，则他可以信任或部分信任 B 的公钥。

只通过一个签名就认为公钥是可信的，这种可信度可能是小了一些。于是 PGP 把用不同私钥签名的公钥收集在一起，发送到公共场合，希望公钥的使用者信任其中一个或多个人的签名，从而间接认证该用户的公钥。

假设用户 D 给他的朋友用户 A 的公钥签名并发布该公钥，或将签名后的公钥回传给

A,这样便可以让 A 通过用户 D 被用户 D 的其他朋友所认可并信任(或部分信任)。与现实中人的交往一样,PGP 会自动根据用户拿到的公钥分析出哪些是朋友介绍来的签名公钥,把它们赋以不同的信任级别,供用户参考,以决定对它们(公钥)的信任程度。也可指定某人具有几层转介公钥的能力,转介后的信任度随着认证的传递而递减。

7.3.4 垃圾邮件防御技术介绍

垃圾邮件存在的原因还有一部分是因为在 SMTP 创造之初,电子邮件传输只是用于学校、政府和军队,因为是一个封闭的系统,所以不存在非法使用和滥用电子邮件的问题。

1990 年以后,因特网被广泛应用于商业用途,但是之前的技术隐患仍然存在。垃圾邮件危害范围逐步增大,反垃圾邮件技术也在不断发展。经过了十几年的发展,新兴的反垃圾邮件技术也层出不穷。

PGP 能够进行邮件的加密传输,验证发送者的身份等。但是,对于垃圾邮件 PGP 却无能为力。概括来说,反垃圾邮件技术的发展经历了如下三个阶段。

(1) 第一阶段主要采用如下技术和手段来抵御垃圾邮件

- 白名单和黑名单技术:黑名单(black list)和白名单(white list)分别是已知的垃圾邮件发送者或可信任的发送者 IP 地址或者邮件地址。现在有很多组织都在试图通过黑名单将那些经常发送垃圾邮件的 IP 地址(甚至 IP 地址范围)收集在一起,以进行垃圾邮件过滤。目前很多邮件接收端都采用了黑白名单的方式来处理垃圾邮件,包括 MUA(mail user agent)和 MTA(mail transfer agent)。当然,黑白名单在 MTA 中使用得更广泛,这样可以有效地减少服务器的负担。
- 简单关键字搜索:该方法一直是对抗垃圾邮件的基本方法。这一功能存在于垃圾邮件成为因特网的首要问题之前,那时它是作为内容过滤的一部分和基于反病毒产品的解决方案和服务的。因为没有文字变化或者上下文对照,所以这种方式只能用作鉴别垃圾邮件的办法之一。但存在很多错误,例如合法邮件经常被误判为垃圾邮件。
- 邮件头测试:信头测试是从收件人、发件人和日期中测试有问题的邮件,如果包含错误形式或者信息便予以阻止。对于不合法的收件人和寄件人,系统将递送一个通告并拒绝递送该信息。这些安全措施用于删除垃圾邮件是非常有效的,可以保证邮件被正确地传送。不管是不是垃圾,只要其中包含了垃圾邮件的信息就会被拦截。
- 简单 DNS 测试:该方法使用 SMTP 协议交换发送者信息的时候,通过查询发送者的因特网域名来验证邮件来源的正确性。比如,查询发送者发送邮件的主机名是否存在(例如通过查询发送者域的 IP 地址和主机名是否相符)。简单的 DNS 测试可以帮助抵御“电子邮件欺骗”。尽管简单 DNS 测试是阻断垃圾邮件的重要手段,但

它只能根据发送者的用户名和地址进行阻断，而不去考察邮件内容是否真的为垃圾邮件，因此它不是严格意义上的一个反垃圾邮件技术。

（2）第二阶段主要采用实时黑名单和电子签名技术

实时黑名单尽管在基于网址和域名上它是一个 DNS 测试，它简单地维护一个发送垃圾邮件的列表以阻止垃圾邮件的继续发送。这种技术对于抵御垃圾邮件有一定的效果，但容易被绕过。例如改变 IP 地址，或者利用第三方的服务器来发送垃圾邮件。同样地，域名会被获取，并被垃圾邮件发送者利用，因而不能完全依赖它来判别垃圾邮件。

电子签名是对于垃圾邮件防御有重要意义的一项技术。其基本思路是如果垃圾邮件以大量的相同信息发送，可以用电子签名技术产生一个伪电子签名来收集和辨别垃圾邮件。如果能够获得充足的垃圾邮件样本，则可以比较有效地判断垃圾邮件。但是这种技术需要及时操作才能达到较好效果。

用鉴别垃圾邮件（签名）和即时黑名单的方法来抵御垃圾邮件的能力有限，垃圾邮件发送者能够轻易地绕过即时黑名单，最好的电子签名技术也无法达到百分之百的正确率。

（3）第三阶段的贝叶斯过滤技术

大约在 2002 年，在因特网和软件行业中出现了一项全新的技术——贝叶斯过滤。贝叶斯过滤利用统计学的方法检测垃圾邮件，基于垃圾邮件中单个词语出现的概率来判定一封邮件是否为垃圾邮件，这是反垃圾邮件技术的一个突破。贝叶斯过滤技术的发展把反垃圾邮件的重点从网络和协议改变为对邮件内容的关注。

简单的贝叶斯过滤使用已经收到的垃圾邮件来训练系统，从而产生一个基于规则评分的系统，来为每封邮件评分。

然而垃圾邮件发送者会不断改变邮件的内容，通常是增加词汇或变种词汇。不断变化中性词语和其他邮件内容及创造变种词汇，使得位于反垃圾邮件系统最后一个步骤的贝叶斯过滤常常被绕过。

2003 年左右，由于新的需求，专门的反垃圾邮件技术开始分离出来，并和一些高科技结合，不断发展起来。

基本上，这些技术执行文件分级使用“非贝叶斯过滤技术”。根据垃圾邮件的变化进行自我更新，目前这一技术正在逐步被使用。

但无论哪一种技术，都无法完全应对多变的垃圾邮件。例如，简单的关键字搜索会产生较多误报。贝叶斯过滤需要经常训练才能适应不断变化的垃圾邮件形式，达到较好的效果。黑名单/白名单因为是硬性地执行拦截或通过命令，使用的时候要非常谨慎。实时黑名单的缺点是它可能产生误报，故应谨慎选择订阅服务。对于 DNS 测试来说，很多反向 DNS 目录未被有效建立，或无法正常建立，这些域发送的邮件将被阻断，造成不可接收的高误报率。

7.4 DNS 安全协议

7.4.1 DNS 脆弱性分析

DNS 为主机提供域名解析服务。DNS 消息采用简单的请求/应答(或称查询/响应)机制:由 DNS 客户端(本章中的 DNS 客户端是指发起域名解析请求的一方,即 DNS 解析器,其本身可以是一个 DNS 服务器)向服务器(指对 DNS 请求进行应答的一方)发出 DNS 查询请求,服务器对其做出应答,一般是把所请求的资源信息发送给客户端。每个 DNS 消息包括一个与之关联的 16 位的 ID 号,服务器根据该 ID 号获取客户端的位置。

DNS 消息结构如图 7.12 所示。一个 DNS 消息由问题区(question count)、回答区(answer count)、权威区(authority count)和附加区(additional count)构成。DNS 数据包的格式及各字段的说明分别如图 7.13 和图 7.14 所示。

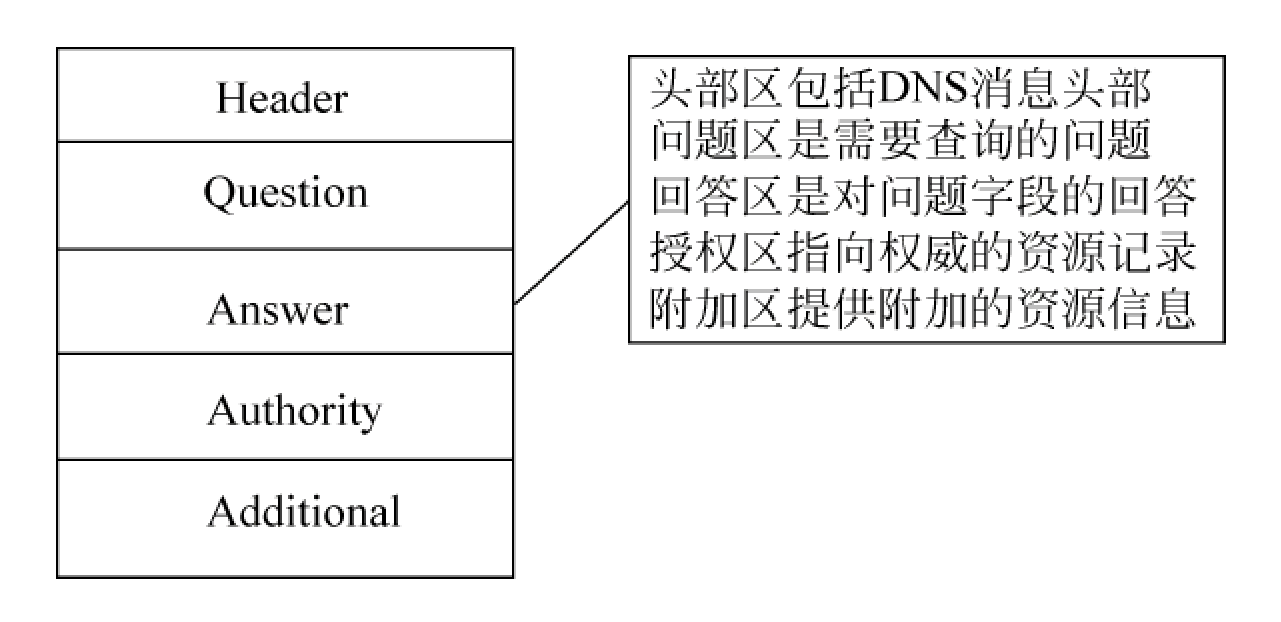


图 7.12 DNS 消息结构

16	21				28		32bit	
ID	Q	Query	A	T	R	V	B	Rcode
Question count	Answer count							
Authority count	Additional count							

图 7.13 DNS 数据包格式

DNS 系统以资源记录(resource record,RR)的形式保存各种资源信息,本章用到的 DNS 的基本资源记录如下。

- A 记录:代表"主机名称"与 IP 地址的对应关系,DNS 使用 A 记录来回答域名查询的 DNS 请求。
- CNAME 记录:代表别名与规范主机名(canonical name)之间的对应关系。
- MX 记录:提供邮件路由信息;提供区域的"邮件交换器(Mail Exchanger)"的主机名及相对应的优先值。当 MTA 要将邮件发送到某个网域时,会优先将邮件交给该

- ID-用于连接查询和答复的 16vbit
- Q-识别查询和答复消息的 1 位字段
- Query-描述消息类型的 4 位数字
  0 标准查询(由姓名到地址)
  1 逆向查询
  2 服务状态请求
- A-命令回答：1 位字段。当设置为 1 时，识别由命令域名服务器作出的答复
- T-切断。1 位字段。当设置为 1 时，表明消息已被切断
- R-1 位字段。由域名服务器设置为 1 请求递归服务
- V-1 位字段。由域名服务器设置表示递归服务的实用性
- B-3 位字段。备用，设置为 0
- Rcode-响应代码，有域名服务器设置的 4 位字段用以识别查询状态
- Question count-16 位字段，用以定义问题部分的登录号
- Answer count-16 位字段，用以定义回答部分的资源记录号
- Authority count-16 位字段，用以定义部门域名服务器的资源记录号
- Additional count-16 位字段，用以定义记录部分的资源记录号

图 7.14　DNS 数据包的字段说明

网域的 MX 主机。同一个网域可能有多个邮件交换器，所以每一个 MX 记录都有一个优先值，供 MTA 作为选择 MX 主机的依据。

- PTR 记录：代表 IP 地址与主机名的对应关系，作用刚好与 A 记录相反。某些网络使用 PTR 记录来检验客户端的主机名称是否可信。
- NS 记录：标记哪些 DNS 服务器可以作为区域的授权服务器。
- SRV 记录：即服务位置资源记录，该记录允许多个服务器提供类似的基于 TCP/IP 的服务，并使用 DNS 查询来定位该服务。

DNS 采用层次化结构，使得主机名可以唯一化。DNS 的结构为反向树结构，由叶节点走向根节点就可以形成一个全资格域名(fully qualified domain name，FQDN)，每个 FQDN 是唯一的。在 DNS 树中，由根到叶给出主机名查询结果，以便于找到属于这台主机的 IP 地址。对于反向映射也有类似的树存在，在树中检索查询 IP 地址的目的是为了找到属于这个 IP 地址的主机名或者 FQDN。这种层次划分域名的方式使得每个主机都可以在其归属的域(或者子域)内有唯一的定义。这样，本地管理员就可以管理(增加、删除或者改动)DNS 主机名和地址。DNS 可以进行主机名本地管理的能力提供了巨大的灵活性和可扩展性。

DNS 的另一个特点是每个区(zone，或称区域、域区等)中包含信息的可用性。除了主服务器外，其余的都成为二级或从服务器，从服务器负责检验主服务器的数据更新，如果检测到有一个数据更新，从服务器就传送域的数据，也就是所谓的区域传输(zone transfer)。每个域都有一个序列号，当主服务器上的域数据更新时，就要调整这个序列号。这种调整使得在服务器上检测到数据更新变得很容易。而能够同时拥有一个以上域备份的能力可以冗

余分配负载，使数据非常可靠。

然而，DNS 高效灵活的设计同时也会引发安全问题。由于 DNS 被设计成一个公共的数据库，在目前的 DNS 协议中对 DNS 域名空间中的信息没有任何访问限制，用户可以随意查询 DNS 中的资源记录，同时攻击者也可能伪造各种 DNS 消息。虽然 BIND（即伯克利 Internet 域名，它是一个由加州大学伯克利分校发展和分发的域名系统执行。BIND 被用在 Internet 上绝大多数的 DNS 服务器中）在后来的版本允许进行某些访问控制，如区域传输等，但是对 DNS 资源记录的查询限制一直排除在 DNS 协议之外。正是由于 DNS 既没有在内部为数据提供安全认证和数据完整性认证，又没有在外部引入任何访问控制机制，使得它存在很多安全漏洞，非常容易遭受攻击。

针对 DNS 的威胁和网络攻击有很多，例如 DNS 欺骗（DNS spoofing）、缓存中毒（cache-poisoning）、拒绝服务（deny of service）、非授权更新（不安全的动态更新）和域名否认存在欺骗等。以下举例说明。

1. DNS 欺骗

域名欺骗是最常见的 DNS 安全问题之一，典型的 DNS 欺骗是一个恶意的攻击者向 DNS 服务器发送欺骗消息，导致受害 DNS 服务器相信并接收了这些消息，同时对系统做了相应的修改。DNS 欺骗可能导致受害 DNS 服务器及其辖区的网络产生严重安全问题。DNS 欺骗包括 DNS 消息劫持、基于名字的攻击及信任服务器背叛等几种方式。

- DNS 消息劫持。利用对数据包（这里指 DNS 请求、应答消息）进行拦截发起攻击。DNS 协议的请求和响应消息依赖于 UDP，因此数据包交换时缺乏认证机制和序列号的控制，发送一个伪造的 DNS 数据包极其简单。如图 7.15 所示，当用户提出 www.wpi.edu 地址解析请求时，正常情况下，DNS 服务器会返回一个正确的 IP 地址，如 130.215.36.202。然而，攻击者很容易捕获到用户请求，并且通过监听服务器和 DNS 客户端（用户）的会话可以猜测到服务器应答消息的 ID 号，如果攻击者先于 DNS 服务器将一个伪造的应答消息（如图 7.15 中的 166.66.66.66）发送给用户，则攻击者就能成功地让用户误以为 www.wpi.edu 的 IP 地址是 166.66.66.66。而

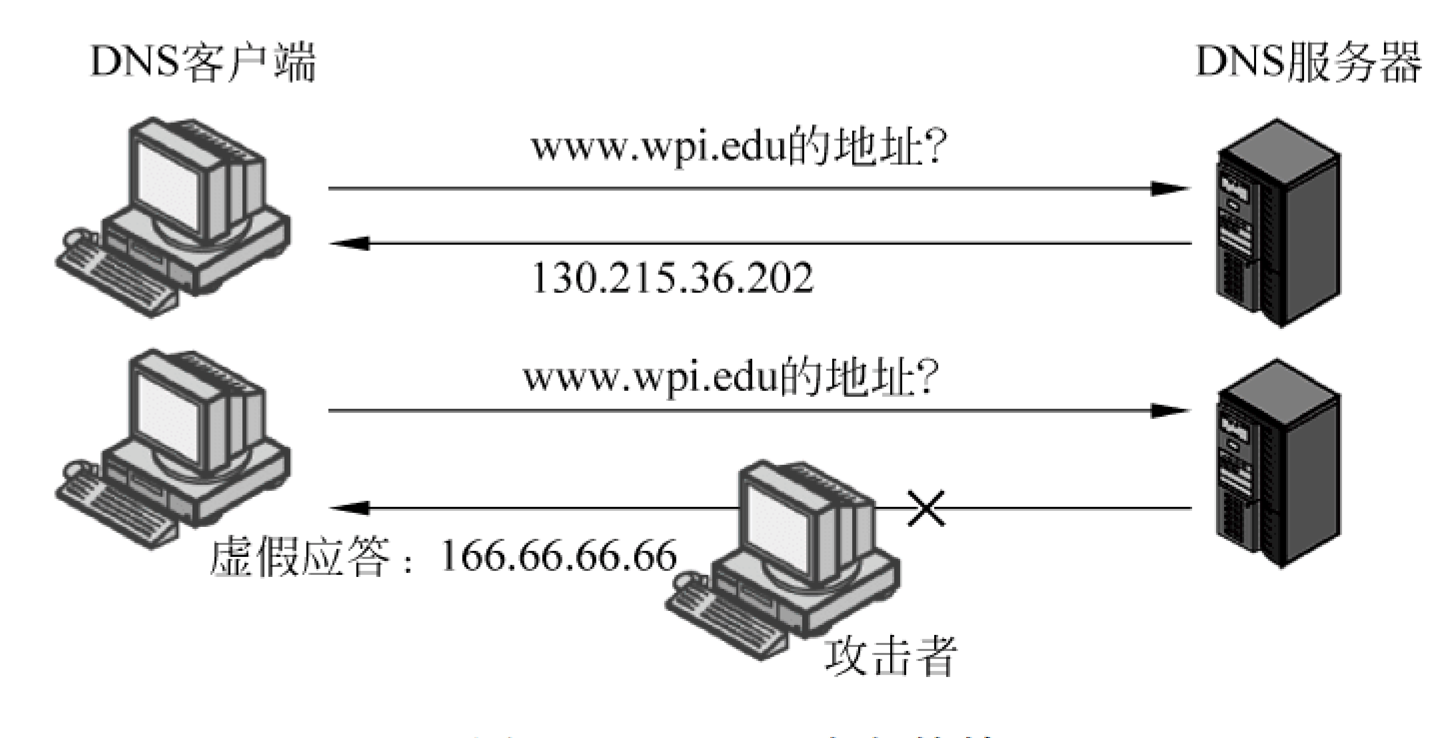


图 7.15　DNS 消息劫持

当正确的 DNS 响应消息到达 DNS 客户端时，却被客户端拒绝了，因为此时已经没有等待应答的请求了。

- 缓存中毒(cache-poisoning)。这是 DNS 面临的一种很普遍的攻击。它利用 DNS 的缓存机制使某个名字服务器在缓存中存入错误或有害的数据。例如，如图 7.16 所示，当某名字服务器 A 收到递归查询请求，而 A 的数据库中没有相应的资源记录时，则它就会将该查询转发给名字服务器 B，B 做出应答，并把应答信息放在应答报文的应答区中，同时 B 在应答消息的附加区中填充一些和查询不太相关的数据。A 接收这条应答报文，而且对附加区中的数据不做任何检查，直接将其存入缓存中。这样使得攻击者可以通过在 B 中存放一些错误的数据而让 A 把错误或有害的数据存放在缓存中，导致 A 的缓存中毒。在这些数据的生存期(time to live，TTL)内，A 又可能会把它们发送给别的服务器，导致更多的服务器缓存中毒。

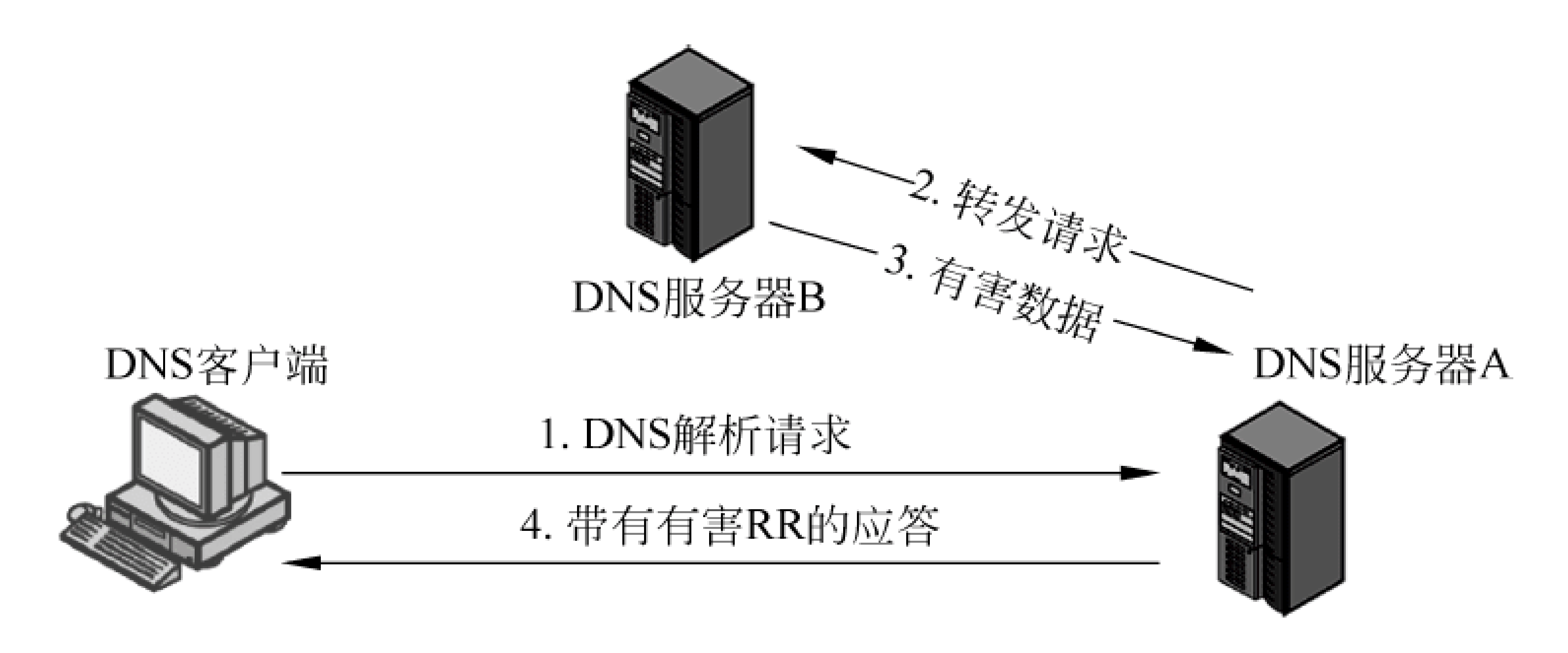


图 7.16 DNS 缓存中毒

假定攻击者利用 DNS 缓存中毒向 ISP 的 DNS 服务器中的 www.microsoft.com 插入受感染的网站，试图访问 Microsoft 网站以下载最新的 Internet Explorer 修补程序的用户会在不知不觉中被重定向到攻击者的网站并下载蠕虫病毒。

- 信任服务器背叛(betrayal by trusted server)。如图 7.17 所示，原本可靠的 DNS 服务器由于受到攻击(也可能是由于其他原因，例如商业目的)变得不再值得信任。除了系统错误和被入侵等原因，有些服务器会出于某种目的返回一些不太符合用户本意的应答消息，可能是一些虚假消息，或者是为了达到商业目的而附加的其他信息。



图 7.17 DNS 信任服务器背叛

移动设备在联网时时常遇到上述问题，它们希望可以随时随地与值得信任的DNS服务器联络。但更多的时候，这种要求达不到，在许多网络环境中，DNS域非常有限，并且并不可靠。甚至在某些极端条件下，DNS端口过滤器和拦截机制阻止了移动终端的交互请求。因此，如果这些问题都是由DNS攻击造成的，就会给用户的安全带来极大的威胁。

2. 拒绝服务攻击(denail of service，DoS)

黑客主要利用一些DNS软件的漏洞，向运行的DNS服务器发送特定的DNS数据包请求，导致DNS服务自动关闭。如果得不到DNS服务，那么整个网络将会陷入混乱。由于网址不能解析为IP地址，用户将无法访问互联网。

3. 分布式拒绝服务攻击(distributed denial of service，DDoS)

类似于DoS，DDoS的目的也是试图使DNS服务器停止服务(指无法提供域名解析服务)。DDoS攻击可以通过使用攻击者控制的几十台或几百台计算机攻击一台主机，使得服务拒绝攻击更难以防范：使服务拒绝攻击更难以通过阻塞单一攻击源主机的数据流，来防范拒绝服务攻击。Syn Flood是针对DNS服务器最常见的分布式拒绝服务攻击。该攻击可以在很短的时间内导致DNS服务器停止服务，危害性很大。

4. 缓存漏洞(buffer overflow)

DNS软件的默认设置是允许主机间进行区域传输(zone transfer)。区域传输主要用于主域名服务器与从域名服务器之间的数据同步，使从域名服务器可以从主域名服务器获得新的数据信息。一旦起用区域传输而不做任何限制，很可能会造成信息泄露，黑客将可以获得整个授权区域内的所有主机的信息，判断主机功能及安全性，并进一步从中发现目标进行攻击。

5. 域名否认存在欺骗

可以认为是DNS欺骗和数据包拦截的一种变形。当客户端向DNS服务器查询的域名不存在时，服务器一般不做一个经过鉴别的应答。此时，攻击者可能截获服务器正常的应答数据包，然后将应答消息中的一个资源记录删除或替换，客户无法检测到这种欺骗，对客户端产生安全威胁。

6. 不安全的动态更新

随着动态主机配置协议DHCP的出现，DHCP客户计算机由DHCP服务器动态分配23地址，使原来手工更新其资源记录(DNS中的A记录和PTR记录)变得难以管理。因此，在RFC 2136中提出了DNS动态更新机制，使得DNS客户端在IP地址或名称出现更改的任何时候可以利用向DNS服务器注册和动态更新其资源记录。尽管DNS动态更新协

议规定只有经过授权的主机才能动态更新服务器的区域文件(zone file),但攻击者可以利用 IP 欺骗伪装成 DNS 服务器信任的主机对区数据进行添加、删除和替换。

可见,DNS 服务的安全漏洞可能被攻击者利用对整个网络实施破坏。因此,加强 DNS 服务的安全性对因特网来说意义重大。

7.4.2 DNS 安全防护策略

目前有一些针对 DNS 的安全防护策略在一定程度上可以缓解 DNS 安全问题。

1. 关闭域名服务器递归查询功能

这种策略关闭递归查询,使 DNS 域名服务器进入被动模式。当它向外部的 DNS 发送查询请求时,只会回答它所授权域的查询请求,而不会缓存任何外部的数据。因此,该方法可以抵御缓存中毒攻击,但同时也降低了 DNS 的域名解析速度和效率。

2. 限制区域传输

在 BIND 配置文件中通过一些设置可以限制允许区域传输的主机,这在一定程度上可以缓解信息泄露。但是,即使封锁整个区域传输也不能从根本上解决 DNS 安全问题,因为攻击者可以利用 DNS 工具自动查询域名空间中的每一个 IP 地址,从而得知哪些 IP 地址还没有分配出去。利用这些闲置的 IP 地址,攻击者可以通过 IP 地址欺骗,伪装成系统内信任网络中的一台主机来完成请求区域传输。

3. SPLIT DNS

该策略采用 SPLIT DNS 技术把 DNS 系统划分为内部和外部两部分。外部 DNS 系统位于公共服务区,负责正常对外解析工作;内部 DNS 系统则专门负责解析内部网络的主机,当内部要查询 Internet 上的域名时,就把查询任务转发到外部 DNS 服务器上,然后由外部 DNS 服务器完成查询任务。把 DNS 系统分成内外两个部分的优势在于,Internet 上其他用户只能看到外部 DNS 系统中的服务器,而看不见内部的服务器,而且只有内外 DNS 服务器之间才能完成 DNS 查询信息的交换,从而保证了系统的安全性,比较有效地防止信息泄露。

4. 及时更新 DNS 服务器软件

因特网上使用最广泛的 DNS 服务器软件是 BIND,它是一个免费软件,其版本在不断更新中,新的版本逐步克服了旧版本的某些漏洞和缺陷。因此使用 BIND 最新版本可以在一定程度上提高 DNS 和网络系统的安全性。但是随着新漏洞的发现,最新版本也同样存在安全隐患。

此外,还有一些辅助手段,例如限制 IP 地址查询、限制进行递归查询的 IP 地址范围等,这些方法和前面的几种方法类似,都是从访问控制和权限限制角度来提高 DNS 系统的安全性。然而,最初设计 DNS 的思想是为公众提供公共的服务,为各种查询提供正确一致的应答,DNS 名字空间中的任何数据都被看成是公共数据。限制这些数据的使用在一定程度上违背了 DNS 的设计初衷,给用户的使用带来不便,同时也不能解决 DNS 面临的基本安全问题,即资源和消息的可信问题。

从上述增强 DNS 安全性的方法中可以看出,这些防范策略仅仅从局部解决 DNS 的安全问题,并没有对 DNS 消息交换过程中的数据提供任何认证和数据完整性检查,攻击者仍可通过各种欺骗手段入侵 DNS 系统,因而没有从根本上解决 DNS 安全问题。有效保护 DNS 系统、提高 DNS 安全性的根本方法是从协议层重新设计域名解析,并在域名解析过程中增加对数据源的认证及完整性验证。

通过上节对 DNS 脆弱性的分析,DNS 安全协议的目标应该包括如下方面。

- 数据源认证:提供 DNS 数据(包括资源、消息)来源的鉴别,例如客户端能够判断 DNS 应答消息是否来自一个真实的 DNS 服务器,从而避免 DNS 欺骗。某些情形下,对于客户端的身份鉴别也是必要的,服务器可以判断一个消息是否的确来源于一个可信的客户端,这样可以抵御 DoS 攻击及防止区域信息泄露。但对于客户端的认证比较困难,带来的额外开销大,适合安全性要求高、小范围实施安全控制的场合。由于区域信息的可靠传输对于 DNS 系统的安全性非常重要,因此可以在区域传输中进行消息鉴别。
- 数据完整性保护:提供 DNS 数据(包括资源、消息)的完整性保护,例如客户端应该能够检测出服务器发出的应答消息是否被篡改过。这样可以有效抵御 DNS 欺骗。

一般不要求对 DNS 请求(查询)和应答消息做加密处理,不对普通查询做过多的限制,因为 DNS 系统是开放的,查询和应答信息本身是可以公开的。

目前,针对 DNS 的安全协议主要有 DNSSEC(DNS security extension)、TSIG(transaction signatures)等。其中 DNSSEC 主要针对 DNS 服务器资源实施保护,通过给资源记录进行签名,可以让 DNS 客户验证服务器的 DNS 应答消息的真伪。TSIG 主要针对 DNS 消息进行鉴别,确保消息来源的真实性。TSIG 允许客户端和服务器之间进行双向身份认证,从而有效防御 DNS 欺骗和拒绝服务攻击(因为服务器可以鉴别客户端的身份),TSIG 特别适合对区域传输过程进行保护。DNSSEC 和 TSIG 可以联合使用,共同提高 DNS 系统的安全性。

7.4.3 DNSSEC 协议概述

为了增强 DNS 协议的安全性,IETF 开发了 DNSSEC。DNSSEC 由 RFC 2535 进行规约,它在现有的 DNS 协议上增添了附加的 DNSSEC 数据类型,以支持对资源的签名和完整

性保护。伯克利 BIND 中实现了 DNSSEC 的部分功能。

在 DNSSEC 中,所有的 DNS 应答报文的内容是经过签名的。客户端可以通过检查消息中的数字签名来鉴别应答消息的来源及其完整性。这种报文内容的实现并不是对整个 DNS 消息进行签名实现的,通常是服务器对其数据库中的资源记录进行签名,然后和对应的 RR 一起保存在数据库中。在客户端发出域名查询时,签名的信息随相应的资源记录一起发送给客户端。客户端通过验证应答消息中的签名信息即可判断收到的消息是否安全可靠。

DNSSEC 可提供如下安全服务。

- 为 DNS 提供消息源认证:即确保应答消息来自可信的服务器,这主要通过对资源记录本身进行签名存储和发布来实现。
- 为数据提供完整性保护:由于对资源记录签名时进行了散列计算,因此可以使接收者能够验证数据在传输过程中是否被篡改过。
- 域名否定存在权威应答:引入新的机制,对域名查询不存在的应答消息提供认证,确认授权域名服务器上不存在所查询的资源记录。并且这种否认应答本身是可信的,从而对域名不存在的 DNS 查询做出权威应答。
- DNS 事务认证:事务鉴别可以对包括 DNS 请求(查询)消息在内的事务处理(通信消息)进行签名和验证,以确认消息本身的合法性。事务签名可以抵御不安全的动态更新等非授权更新,以及数据包篡改类型的攻击。这种保护对于区域传输尤为重要。为了提高 DNSSEC 对消息及事务处理本身的鉴别能力,可以使用 TSIG 机制,也可以使用 IETF RFC 2931 定义的"DNS 请求和事务认证"机制,称为 SIG(0)。前者使用对称密码,后者使用公钥密码。目前,TSIG 比 SIG(0)的应用更广泛。

DNSSEC 的基本功能是提供对其资源记录的签名保护,即主要提供上述前三个安全服务。DNSSEC 采用公开密码体制实现服务器消息的数据源认证和消息数据的完整性保护。首先,为资源记录计算一个消息散列值(或消息摘要),然后使用服务器的私钥对该散列值进行非对称加密,形成签名。为此,要将一个 DNS 的区域转变为一个安全的 DNSSEC 区域,或签名区域,管理员需要为这个区域生成一个公钥/私钥对。私钥用于对资源记录进行签名,然后妥善保存;公钥采用某种方式发布给客户端,以便进行签名验证。

为了实现这些安全服务,DNSSEC 还需要进行密钥管理和信任关系的维护(见 7.4.4 节和 7.4.5 节)。

1. DNSSEC 的新增记录

为了提供资源签名和完整性保护,DNSSEC 新增了如下基本安全资源记录。

- SIG。DNSSEC 中,SIG 作为扩展的签名资源记录(SIG RR)存在,用于存放对应资源记录的签名信息。区域所有者使用自己的私钥对各个资源记录进行签名,形成相应的 SIG RR。当 DNS 客户端向 DNS 服务器发出查询请求时,服务器在消息的附

加区域中加入该 RR 对应的 SIG,即把资源记录的签名附加在应答数据包中。客户端负责鉴别数据的真伪和完整性。

- KEY。增加密钥资源记录(KEY RR),用来存放和发布服务器的公钥。一般,每个区域拥有一对私钥及其对应的公钥。私钥用于对资源记录进行签名,公钥需要发布到客户端,客户端使用它进行签名验证。公钥的拥有者可以是一个区域、主机或某个用户。KEY RR 本身需要进行签名(以防止假冒),然后作为 DNS 资源记录的一种随 DNS 应答消息一起发送给客户端。
- NXT。设立 NXT 类型记录(NXT RR),以可靠地否认一个域名的存在(对于区域中不存在的域名用 NXT RR),从而抵御攻击者利用 DNS 系统中域名查询不存在的漏洞进行的攻击。DNSSEC 中,DNS 服务器对一个不存在的域名将返回 NXT RR 和它的签名。NXT RR 是经过排序的,域名中与这个被查询的域名最接近的下一个域名的 RR。排序的方法一般是把域名看成一系列经过划分的字串,从最高字串排起,若最高字串相同,再排次高字串,依次类推。当查询的域名排在最后的域名之后时,DNS 服务器将返回排在第一位的域名,即把所有域名看成一个循环的队列。当客户端查询的 DNS 资源不存在时,安全的 DNS 服务器至少要返回一个已经签名的 NXT RR,供客户端验证。而在现有的非安全 DNS 服务器中,当查询的域名不存在时,服务器不会自动返回 NXT RR,只有当显式请求时才会返回 NXT RR。
- DS。即委托签名者(delegation signer),是一个指向 DNSSEC 信任链的指针。

图 7.18 中给出了普通 DNS 记录格式的示例,图 7.19 中给出了使用 DNSSEC 后的记录格式例子,可以看出在 DNSSEC 中每条记录都被进行了签名保护。

```
ct.nl.          IN     SOA     ns.xtdnet.nl. hostmaster.ct.nl. (
                    2003021618 ; Serial
                       28800   ; Refresh
                       7200    ; Retry
                       604800  ; Expire
                       3600)   ; Minimum
ct.nl.          IN     NS      ns.xtdnet.nl.
ct.nl.          IN     NS      ns1.xtdnet.nl.
www.ct.nl.      IN     CNAME   www.fnl.nl.
ct.nl.          IN     MX      10 cable.fnl.nl.
ct.nl.          IN     MX      30 smtp.xtdnet.nl.
```

图 7.18 普通 DNS 记录

2. 新增记录的数据结构

引入新增的 RR 后,DNSSEC 需要采用某种数据结构(DNS 中称为 RDATA)来描述该资源记录。

(1) KEY 资源记录

KEY 资源记录的 RDATA 如图 7.20 所示。其中,公钥字段代表某个实体的公钥,其他字段说明如下。

```
3600    SIG    MX 5 2 3600 20030320173018 (
               20030218173018 35861 ct.nl.
               S1261RNyGntq+PCZ65xe )
3600    KEY    256 3 5 (
               XDXiK4oxgbcdJx51mDsl
               ) ; key id = 57410
3600    KEY    256 3 5 (
               8a+qDT/20al/y2x5sWQ+mgk=
               ) ; key id = 35861
3600    SIG    KEY 5 2 3600 20030320173018 (
               20030218173018 35861 ct.nl.
               CthM1Kdv1IY1528jKL5P )
3600    SIG    KEY 5 2 3600 20030320173018 (
               20030218173018 57410 ct.nl.
               dRZ+MF+7+Zt0aphQiw== )
3600    NXT    www.ct.nl.NS SOA MX SIG KEY NXT
```

图 7.19　DNSSEC 记录

标志	协议	算法
公钥		

图 7.20　对 KEY 资源的描述(RDATA)

- 标志(flag):指示 KEY 资源记录所代表的是何种实体,即指明公钥拥有者的身份(如区域、主机或某个用户)和其他一些附加属性。
- 协议(protocol):现在只代表 DNS,但 KEY 资源记录将来还可用于别的因特网协议。因为 KEY 资源记录可以作为一种通用的公钥分配方案,不应仅局限于 DNSSEC。
- 算法(algorithm):指明生成和验证签名时使用的加密算法,可选用 RSA,也可以是其他一些公开密钥算法或某种内部商定的算法,此时 Public Key 域的格式与加密算法有关。注意,对于一些特殊的加密算法,Public Key 域(字段)还可以有它的子域。

(2) SIG 资源记录

SIG 资源记录的 RDATA 如图 7.21 所示。

覆盖类型	算法	标签
原始 TTL 值		
签名失效		
签名时间		
密钥印迹	签名者的名字	
签名者的名字		
签名		

图 7.21　SIG 资源的描述(RDATA)

其中名字段的说明如下。

- 覆盖类型(type covered):指定对哪种类型的 RR 进行加密。
- 算法(algorithm):与 Key RR 中的字段相同。
- 标签(labels):域名中的字段数,用以快速决定域名是否使用了通配符。

• 原始 TTL 值(original TTL):表明对资源签名时的 TTL 值,因为报文在传输过程中 TTL 值会改变,所以必须保存这个值,当验证时以此值替换改变后的 TTL。
• 签名失效(signature expiration):指明该签名的失效期。
• 签名者的名字(signer's name):指明对该 RR 进行签名的签名者的名字。
• 签名时间(time signed):指明对该 RR 进行签名的时间。
• 密钥印迹(key footprint):当有多个 Key 可选时,用该字段能快速地选出一种。
• 签名(signature):是加密后的密文(签名)。

DNSSEC 中的签名和验证签名会产生额外的开销,从而影响网络和服务器的性能。如果签名的数据量很大,就会加重 DNS 对 Internet 骨干网及一些非骨干连接的负担。

此外,如果使用 DNSSEC,还需要对原有的 DNS 软件进行改造。DNSSEC 软件自身还需要完善和更新,需要进行实际操作和测试。目前,DNSSEC 还没有和 Internet 的迅速发展同步,仍在不断发展之中。

7.4.4 DNSSEC 密钥管理

基本上 DNSSEC 协议一般只提供数字签名服务,因此 DNSSEC 系统中需要管理的密钥主要是公钥/私钥对。私钥用来为区数据签名(或为另一个公钥签名),公钥对签名数据进行验证,完成数据源的真实性和完整性鉴别。

1. 密钥生成

这里的密钥指公钥、私钥对。在 DNSSEC 中,每个保护区(secure zone)都有对应的密钥对。对于公开密钥算法而言(如 RSA 算法),生成密钥的关键是大素数的生成。在一般非密码系统的应用场合,只要求产生的随机数呈现平衡的、等概率分布即可,并不要求它的不可预测性。而在密码系统中,特别是在密钥生成技术中,不可预测性成为产生随机性的一条根本原则。一般而言,重复地从某一个固定区间随机选取一个整数 n,并对 n 进行素性检测,即可得到所需的大素数。

DNSSEC 中密钥对生成可以使用工具,比如 OpenSSL。根据一些非对称密钥算法,随机生成公钥、私钥对,生成的参数包括使用的加密算法(如 RSA、DSA、MD5 和 DH 等)、密钥长度和密钥所有者等。

通常一个区域需要一对公钥/私钥。如果一个安全的解析器可以得到一个可信的公钥,它就可以验证从该区发送的、经过签名的数据是否可靠。

2. 密钥的分发和管理

这里的密钥指公钥。对于公钥的分发,DNSSEC 实现起来非常方便,它不需要采用专门的机制提供公共手段分发大量的公钥(除了系统的根公钥需要一个程序定期从根服务器

上下载以外，见下节）。如前所述，公钥被保存在DNS资源记录中的KEY RR中，并随所请求的DNS数据信息一起封装在DNS应答消息中传送至客户端。对于每个域名请求，根据其请求包的某个标识，如果判断出请求方已具备DNSSEC功能，则响应包中不仅有RR数据，还包含公钥数据和对应的签名数据。

客户端使用公钥进行签名验证之前，首先要确保公钥本身是可信的。因此，对于公钥的管理，DNSSEC主要是要解决如何证明公钥是真实、可信的问题，DNSSEC可以通过信任链的方式来实现公钥信任机制：KEY RR中的公钥本身是经过签名的，因此需要另一个公钥来鉴别当前公钥的真实性，依次类推，一系列公钥构成了公钥信任链，最高级别的公钥称为根公钥(root key)，它应该被系统中的所有成员信任。

3. 密钥存储

DNSSEC的安全体系构建在签名和验证签名的过程上，因此私钥本身的安全可靠是系统安全的基础。每个区需要一个私钥对该区内的数据进行签名，区数据的安全性取决于区的私钥的保护。

DNSSEC中，区的私钥离线存放在某个安全的地方——通常在域名服务器文件系统的一个文件里。该文件只有运行密钥生成程序者才有访问和读写权限。区的公钥则可以存放在DNS域名信息的配置文件中。除了公钥本身，还需要放一些与公钥相关的信息，如公钥长度、加密算法等。

4. 密钥的注册

密钥对的公钥部分需要由父区来签名以确认其真实可信。注册的目的在于确认其身份和公钥真实有效，密钥经过注册后可以用来对所获得的签名数据信息进行鉴别。

5. 密钥的更改

在公钥密码机制下，因为各种原因密钥对(公钥、私钥对)必须要及时更换。密钥的更换其实就是一次对区记录进行重新签名和重新认证的过程。由于DNS采用缓存结构，新密钥的生成不一定能为所有解析器立即接受。旧密钥及其签名记录可以和新的签名共存一段时间，直到旧公钥从远端服务器的缓存中清除。

6. 分层的密钥结构

DNSSEC采用分层的密钥结构，即子区和父区的密钥分层存储和签名。根据签名对象的不同，又分为两种密钥结构。RFC 2535中定义的基本密钥结构如图7.22(a)所示。即子区私钥对子区记录数据进行签名，而子区公钥由父区私钥签名认证。

RFC 4641(DNSSEC操作实践)中描述了另一种密钥结构，如图7.22(b)所示。这种结构把每个区域的公钥和私钥都分成两种：一种命名为ZSK(zone signing keys)，是加密区记

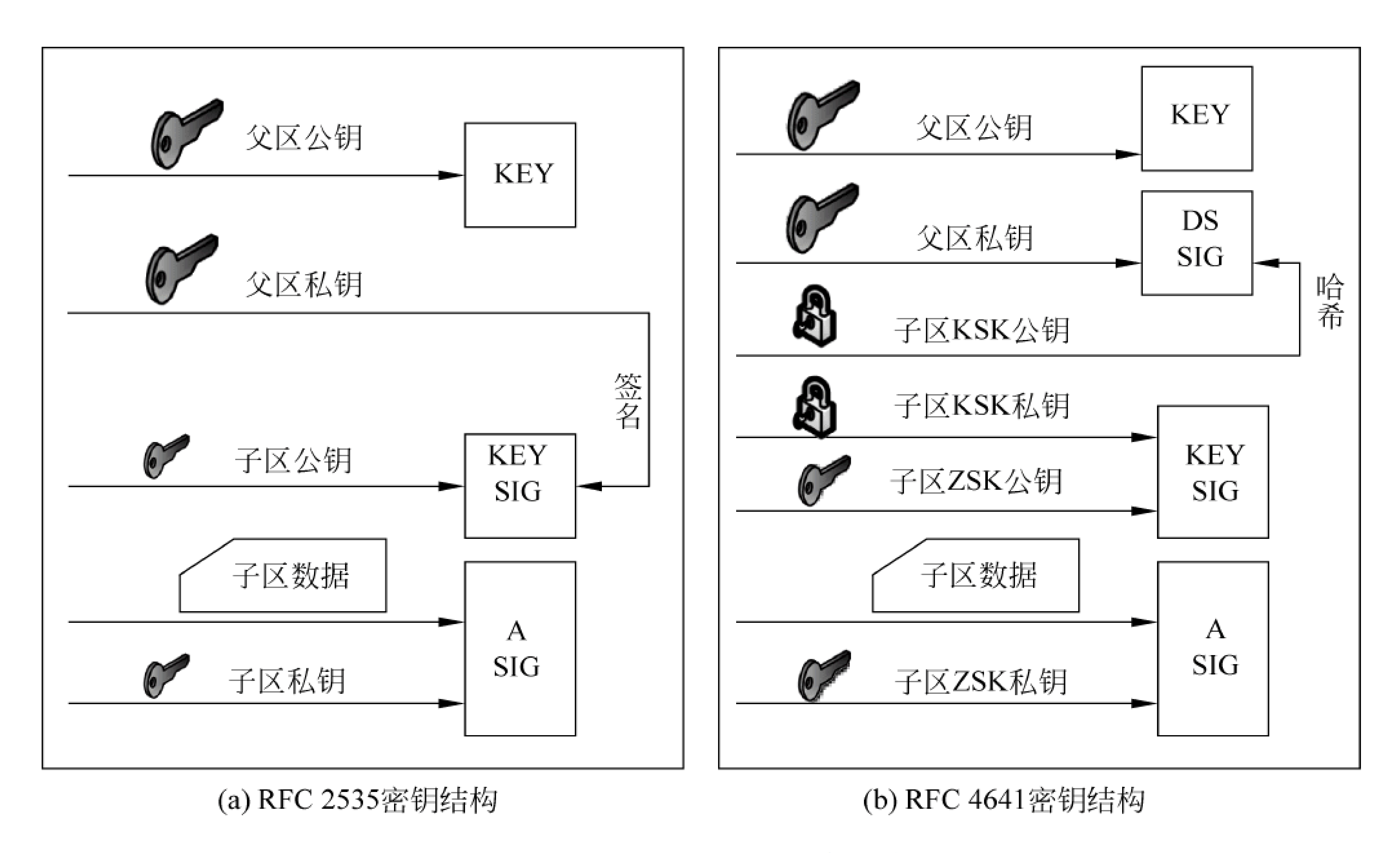


图 7.22　DNSSEC 的密钥结构

录的密钥；另一种命名为 KSK（key signing keys），即加密密钥的密钥，这两种密钥分别有公钥和私钥之分。签名按如下方式进行：子区 ZSK 私钥为子区记录签名，子区 ZSK 公钥验证其签名；子区 KSK 私钥用来为子区 ZSK 公钥签名，子区 KSK 公钥用来验证其签名；子区 KSK 公钥又由父区私钥进行签名。这样设计的目的是方便密钥的撤销，增强区数据的安全性，但同时也增加了协议的复杂性。

7.4.5　DNSSEC 签名验证及公钥信任机制

为了有效进行签名验证，DNSSEC 必须建立一个有效的公钥信任机制。目前 DNSSEC 的信任机制主要使用的是委托信任链机制。

DNSSEC 基于非对称密码算法，DNS 消息认证需要用响应者发送的公钥对接收到的消息进行数据源鉴别，同时验证数据的完整性。因此，公钥是否可信对整个 DNS 系统至关重要。为了保证公钥的正确性，DNSSEC 要求建立一个信任链表。其核心思想是父区对子区的公钥进行数字签名，即由父区来验证子区公钥的可靠性，以保证 DNS 客户端（本身可能是一个 DNS 服务器，担当 DNS 解析器的角色）能够获得可信的公钥。例如．com、．org 等区的解密公钥由根域名服务器掌管的私钥进行加密签名，x．com 的公钥由．com 区来签名，依次类推。公钥信任链一般起始于域名解析器信任的公钥。该点称作安全入口点，根据 RFC 2535 定义，该公钥应该是根域名服务器掌管的公钥，所以安全入口点实际上就是根域名服务器。此公钥即为根公钥（root key），它可以通过网站公开发布，让所有的域名服务器、主机都预

先静态配置在自己的机器上。

实现时，DNSSEC 使用委托代理资源记录（DS RR）指向给某个实体签名的密钥（通常是父区的密钥），即指明信任链中下一个可信的密钥，由 DS 指向的密钥的签名信息应该是可信的。

图 7.23 描述了一个 DNSSEC 查询和应答的具体实例，描述了 DNSSEC 公钥验证的信任链机制。

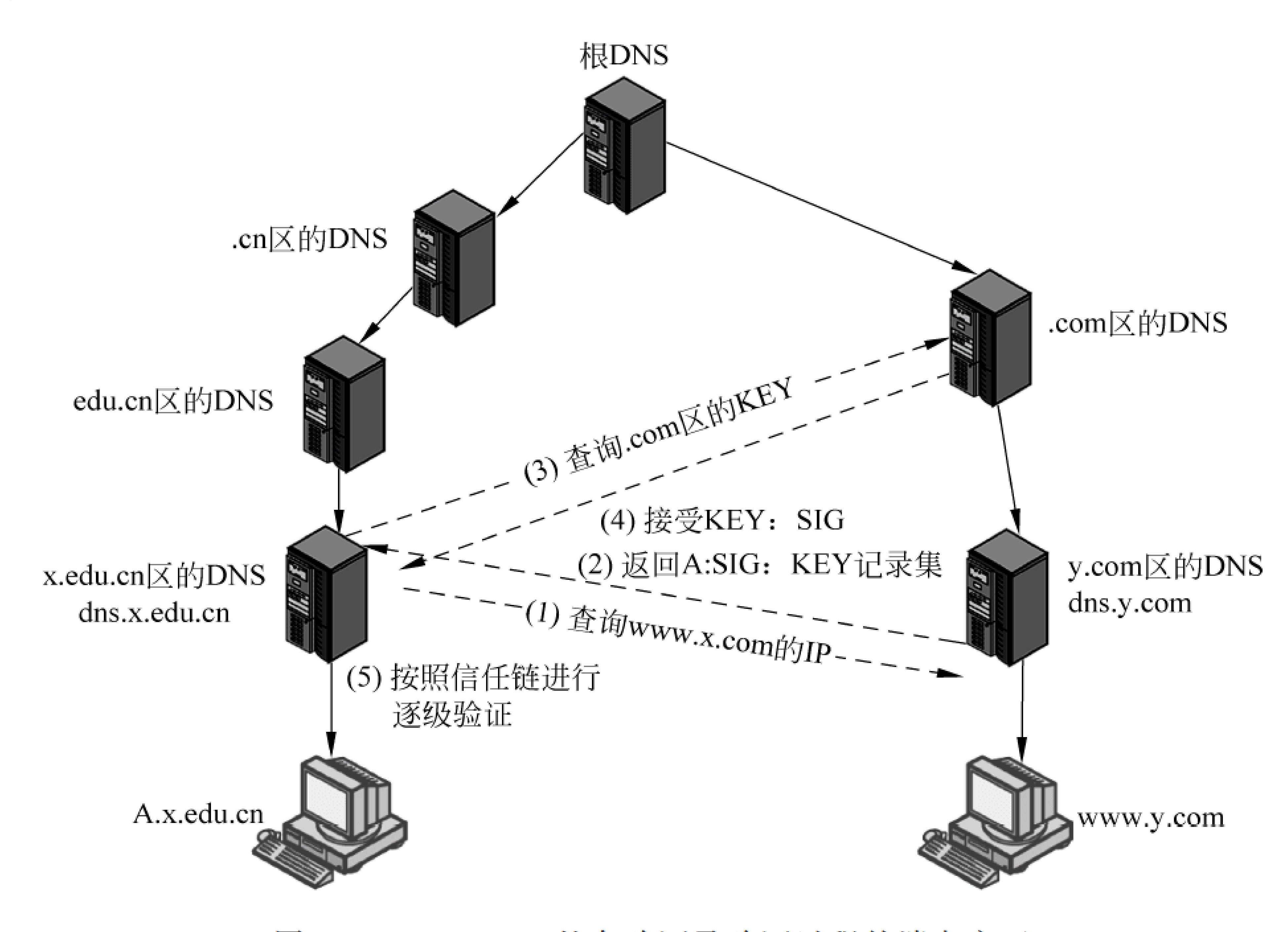


图 7.23 DNSSEC 签名验证及验证过程的消息交互

初始时，所有的域名服务器都配置好了根公钥，随后，客户机 A. x. edu. cn 向 DNS 系统查询域名为 www. y. com 的主机的 IP 地址。该请求被本地域名服务器 dns. x. edu. cn（它是 x. edu. cn 区的域名解析器）接收，并利用 DNSSEC 的委托信任链机制做如下处理。

① dns. x. edu. cn 中没有该主机的资源记录，于是它向 y. com 区的权威 DNS 服务器 dns. y. com 发起 DNS 请求。

② dns. y. com 作出 DNS 应答，应答消息包括主机 www. y. com 的 A 记录（A 记录在 DNS 系统中为标识主机地址的资源记录）、A 记录的签名记录和 y. com 区的公钥（该公钥由. com 区的私钥进行签名）。

③ dns. x. edu. cn 为了验证 y. com 区公钥的有效性，向 y. com 的父区即. com 区发出公钥请求消息。

④ . com 区的域名服务器向 dns. x. edu. cn 作出应答，应答消息中包含. com 区的公钥及公钥的签名记录。

⑤ dns. x. edu. cn 首先利用 Root Key 验证. com 的公钥的合法性，确认后再利用. com 的公钥验证 y. com 区公钥的合法性。最后利用 y. com 对 A 记录的签名进行验证，确认该记录合法，且内容完整后接收该记录，并向客户端主机 A. x. edu. cn 发出域名解析的应答消息。签名验证过程中如果其中有一个区的公钥认证未通过，就认为这个区是不安全的。

其中，步骤①中的消息格式如下。

Question:
Qname=www. y. com
Qtype=A;
RDATA=?

步骤②中的消息格式如下。

问题区：	1. www. y. com	A	?;
应答区：	2. www. y. com	A	10. 10. 10. 1;
	3. www. y. com	SIG	[RDATA info];
权威区：	4. y. com	NS	dos. y. com
	5. y. com	SIG	[RDATA info];
附加区：	6. dns. y. com	A	10. 10. 5. 3
	7. dns. y. com	SIG	[RDATA info];
	8. y. com	KEY	[RDATA info];

第 1 条记录是原始查询，在 DNSSEC 的应答报文中，不仅包含了问题的回答，还包括了原始的查询。第 7 条记录即是对查询和应答报文的连接进行的数字签名。在 DNSSEC 中这条资源记录是应答中必须包括的。通过这两条记录 DNSSEC 就可以提供事务和查询认证，保证应答的数据的确是对原始查询的应答，同时也确实来自于被查询的服务器。

第 2、3 条记录分别是 www. y. com 的 A 记录和这条 A 记录的签名记录。

权威区中用来存放 NS 记录及它的签名记录，以说明 y. com 区的授权，该例中 y. com 区授权 dns. y. com 为该区 DNS 权威服务器。

附加区中 6、7 条记录分别是 dns. y. com 的 A 记录和它的签名；第 8 条记录存放了 y. com 区的解密公钥。解析器获取这个公钥并通过上面描述的公钥信任链对此公钥认证无误后，就对 3、5、7 这三条 SIG 记录进行签名验证，如果验证不通过，则丢弃该报文。

7.4.6 TSIG 和 TKEY

如前所述，DNSSEC 着重保护 DNS 资源记录。为了增强传统 DNSSEC 的事物鉴别能力，即保护 DNS 通信消息的安全传输，IETF 开发了事务签名 TSIG。TSIG 由 RFC 2845 进

行规约。TSIG 使用共享密码对 DNS 消息实施鉴别，这尤其适用于保护 DNS 解析的响应和更新、保护区域数据传输。对消息的保护可以是双向的，因此，TSIG 中，客户端和服务器的消息均可以进行认证和鉴别。

如图 7.24 所示，当配置了 TSIG 后，DNS 消息将会增加一个 TSIG 记录数据项，该数据项对 DNS 消息进行签名，签名信息附加在 DNS 消息的尾部，提供消息源鉴别和完整性保护。

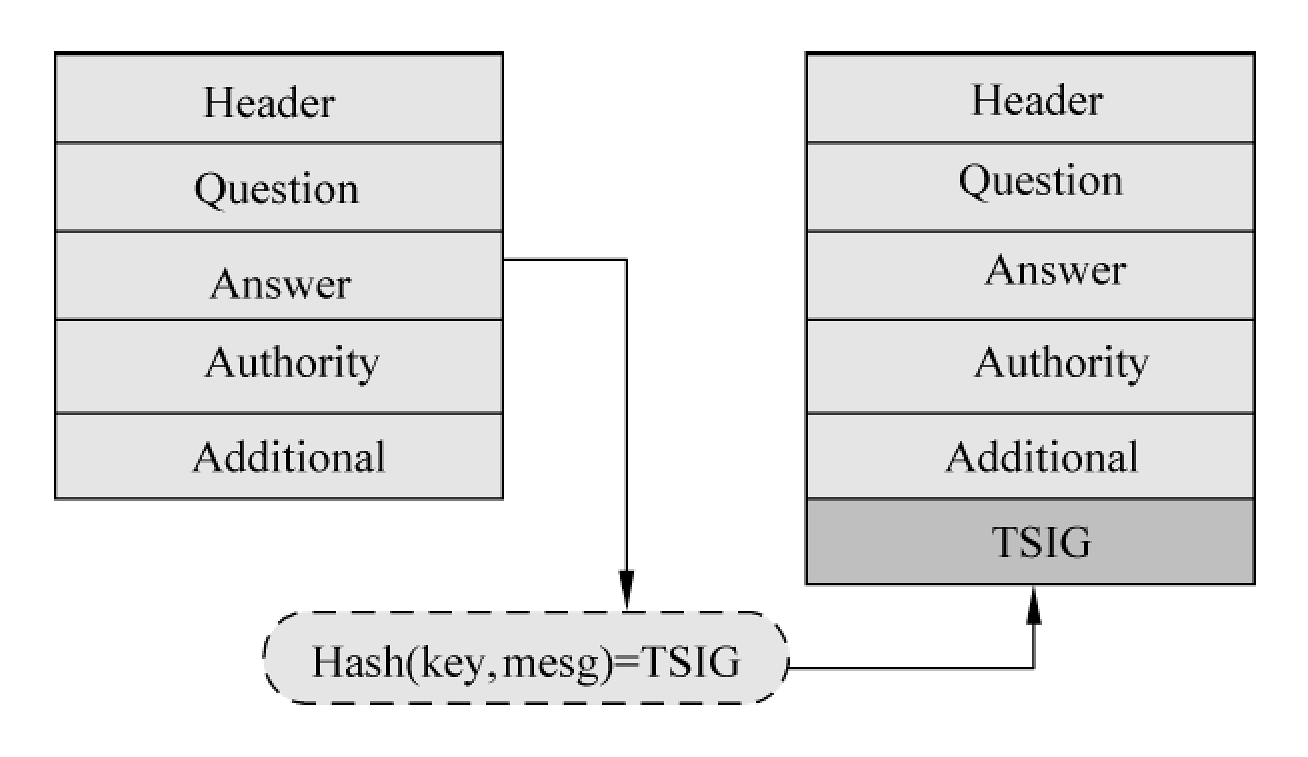


图 7.24　TSIG 对 DNS 消息的封装

签名消息通过 HMAC 算法实现，即 TSIG＝Hash(key，message)。因此，通信双方需要一对共享密钥。TSIG 中的共享密钥可以在一台主机上产生，然后使用安全的方式分发给对等实体。也可以通过协议自动产生共享密钥。

TSIG 中使用 TKEY 为 DNS 客户端和服务器之间自动生成共享密钥，即用来自动产生 TSIG 中的加密密钥。TKEY 消息交换过程本身必须使用签名信息，可以使用 TSIG 或者 SIG(0)签名。TKEY 也被用来删除先前使用过的密钥。初始时，客户机向服务器发送一个签名的 TKEY 请求消息(包含一个适当的密钥)，密钥发送到一个理解 TKEY 的服务器。服务器进行应答，如果成功，它会包含一个 TKEY 记录和一个适当的密钥。经过这个交换后，双方都有足够信息来计算共享密钥，细节的过程依赖于 TKEY 的模式选择。可以采用 Diffie-Hellman TKEY 模式交换密钥，以得到通信双方产生 TSIG 签名的共享密钥。图 7.25 中给出了一个 DNS 通信双方使用 TKEY 协商产生共享密钥的例子。

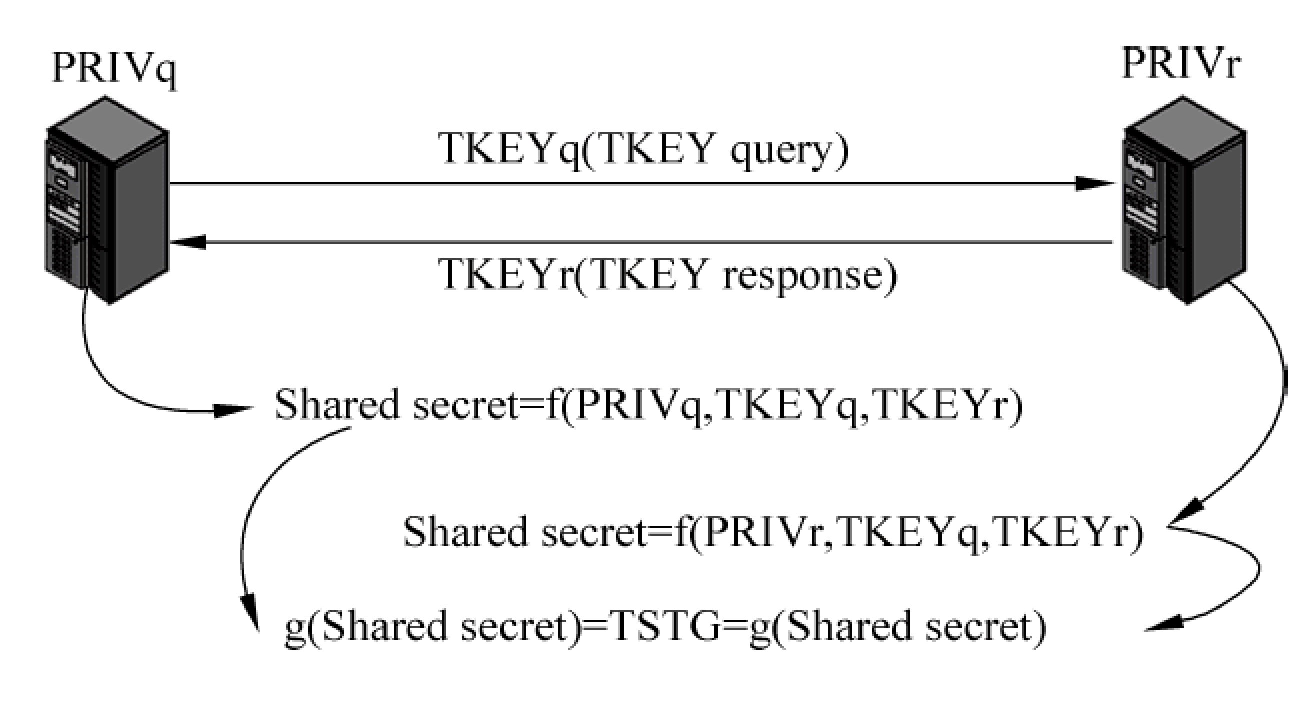


图 7.25　TKEY 产生共享密钥

TSIG 中加入了时间戳,以抵御重放攻击,这要求 DNS 客户端维护一个精确的时钟。TSIG 协议的对等实体可以通过网络时间协议(network time protocol,NTP)来获取精确的时间信息。

TSIG 的缺点是共享密钥必须在线传递,或通过某种私有渠道分发给需要进行消息鉴别的实体,这给使用者带来了不便,也限制了其使用的范围和规模。

目前,TSIG 和 DNSSEC 已被共同集成到 BIND 中,TSIG 可以和 DNSSEC 一起使用,也可以单独使用。

7.5 SNMP 安全协议

7.5.1 SNMP 及其安全性概述

简单网络管理协议(simple network management protocol,SNMP)是目前 TCP/IP 网络中应用最为广泛的网络管理协议。SNMP 的发展经历了几个重要阶段。1990 年 5 月,RFC 1157 定义了 SNMP 的第一个版本 SNMPv1,它提供了一种监控和管理计算网络的系统方法,该方法简单易行,获得了许多厂商的支持,也得到了广泛的应用,并成为网络管理事实上的标准。

SNMP 在 20 世纪 90 年代初得到了迅速发展,同时也暴露出其性能上的明显不足,如功能简单、难以实现大量的数据传输,缺少身份验证和加密等安全机制等。因此 1993 年发布了 SNMPv2,它在 SNMPv1 的基础上进行了功能性改进:支持分布式网络管理;扩展了数据类型;可以实现大量数据的同时传输;丰富了故障处理能力。

然而 SNMPv2 并没有有效解决 SNMP 协议面临的安全问题,网络管理协议和通信过程面临着假冒、信息篡改、报文延迟、重播和报文序列的修改、信息暴露等安全威胁,这些安全问题阻碍了 SNMP 的应用和发展。

为增强网络管理协议的安全性,IETF SNMPv2 工作组于 1998 年发布了 RFC 2271～RFC 2275,正式形成了 SNMPv3。SNMPv3 实现了 SNMPv2 未能实现的几个目标,它采用基于用户的安全模式(user-based security mode,USM),在安全性和管理机制等方面对 SNMPv2 进行了扩展。

SNMP 由一系列协议和规范组成,它们提供了一种从网络设备中收集网络管理信息的方法。SNMP 网络管理模型采用 Manager/Agent 结构,即管理者/代理模式。

如图 7.26 所示,一个支持 SNMP 的网络管理系统包含 4 个基本要素:管理者(manager)、代理(agent)、管理信息库(MIB)及 SNMP 消息(SNMP message)。其中,MIB 是一个存放被管设备(如 PCs、工作站、服务器、网桥和路由器等)所维护的全部被管理对象的数据库。在 SNMP 管理体系中,被管设备的各种数据变量被抽象为不同的被管对象,全部被管对象

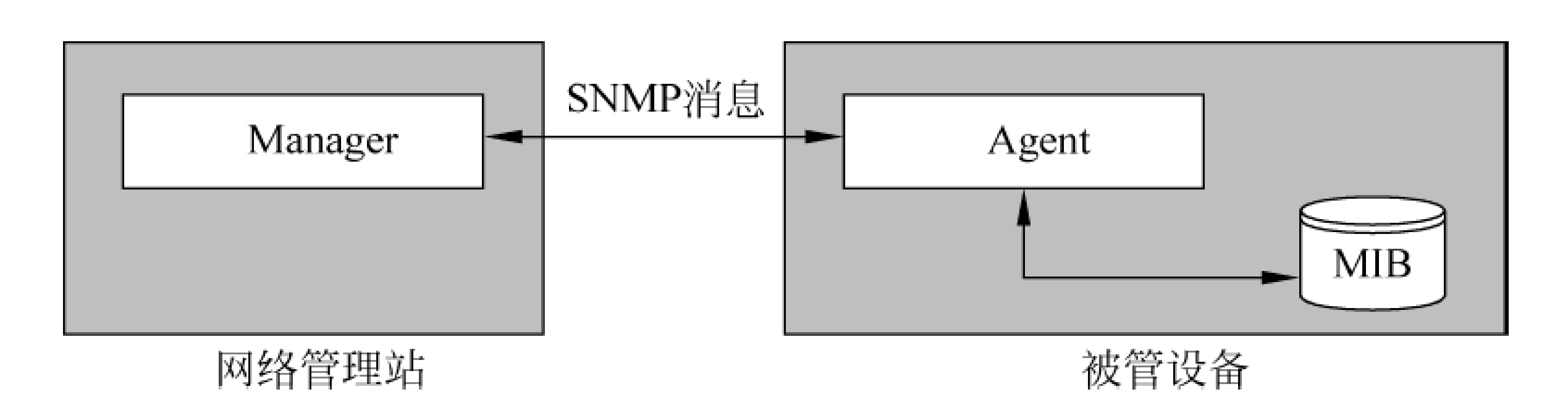


图 7.26 SNMP 网络管理系统的构成

的集合被组织为 MIB。Agent 是安装在被管理设备中的软件模块，负责维护本地 MIB 及响应 Manage 发来的消息，此外它还可以主动向 Manager 通报重要事件的发生。Manager 是系统的管理者，安装在网络管理站上，它可以读取和设置 Agent 所维护的 MIB 中被管对象的值，并和 Agent 之间进行各种信息交互，交互信息封装在 SMTP 消息中发送。即 Manager 与 Agent 之间，Manager 与 Manage 之间通过 SNMP 消息交互信息，SNMP 消息使用 UDP 端口 161/162 进行传输。

由于因特网是一个开放平台，传统 SNMP 协议中的消息传输又是明文的，这使得在网络管理中使用 SNMP 时，可能面临严重的安全威胁，包括如下方面。

- 篡改：非授权实体篡改正在传递中的由授权实体产生的 SNMP 消息，以此执行未被授权的管理操作。
- 伪装：非授权用户假冒授权用户来进行未被授权的管理操作。
- 滞延和重传（重放）：管理消息在传输过程中被故意延迟或重复发送。
- 窃听：消息在传输过程中非授权用户对其进行复制，通过消息副本来获取消息中的信息。

为提高 SNMP 的安全性，SNMPv3 需考虑提供以下几种安全服务。

- 数据完整性：保证数据在传输过程中未被修改。
- 数据机密性：必要时，对数据进行加密使其内容不被泄露。
- 数据来源认证：实现对接收数据的发送者的身份认证。
- 消息及时性保护：保证只处理在有效时间内接收到的消息。
- 访问控制：通过对合法用户的授权，实现控制访问者对不同被管资源的访问权限。

7.5.2 SNMPv3 的体系结构

1. SNMPv3 体系架构

SNMPv3 提出了一个新的 SNMP 体系架构，这个体系架构为各种基于 SNMP 的管理系统提供了一个通用的实现模型。SNMPv3 将网络看成由许多分散的相互作用的 SNMP 实体构成，这些实体或者是 Agent，或是 Manager，或者是两者的结合。通过 SNMP 实体之间的相互作用来实现对网络及其资源的检测和控制。其中，每个 SNMP 实体由一个 SNMP

引擎和若干个 SNMP 应用构成,如图 7.27 所示。在一个管理域中,SnmpEngineID 唯一标识一个 SNMP 引擎。不同管理域中的两个 SNMP 引擎允许有相同的 SnmpEngineID。由于 SNMP 实体和 SNMP 引擎是一一对应的,所以 SnmpEngineID 也用来唯一标识一个 SNMP 实体。

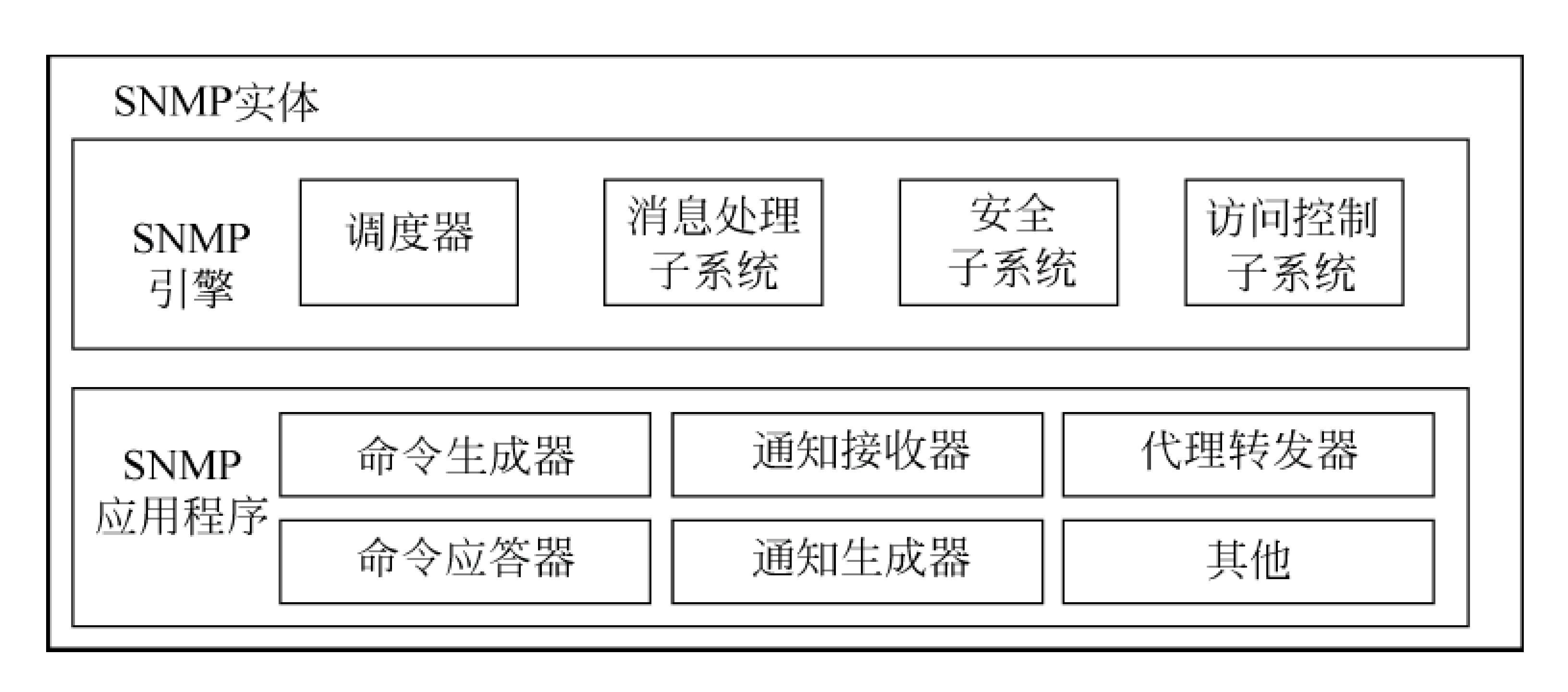


图 7.27 SNMPv3 体系架构

SNMP 引擎为 SNMP 应用提供消息的接收和发送、消息认证、加密和访问控制等服务,主要由以下几部分组成。

- 调度器(dispatcher):是 SNMP 引擎的关键部件,每个引擎只有一个调度器,它负责 SNMP 消息的接收和发送。调度器和消息处理子程序协作完成消息的进一步处理,负责发送 PUD 到 SNMP 应用程序,同时通过底层网络协议协作完成消息的传输。调度器能够为不同版本的消息处理子系统分派任务,并为不同的应用提供发送和接收 PDU 的服务,其功能包括向网络发送或从网络接收 SNMP 消息;确定消息的版本;与相应的消息处理模型交互;为应用提供抽象接口,向其传递 PDU 和接收其欲向其他实体发送的 PDU 等。
- 消息处理子系统(message process subsystem):负责准备要发送的 SNMP 消息及从收到的消息中抽取数据。SNMPv3 充分考虑到和之前版本的兼容性,消息处理子系统包括多个消息处理模块,每个消息处理模块定义一个特定版本的 SNMP 消息的格式,以对不同的消息格式进行不同的处理。
- 安全子系统(security subsystem):负责提供消息的认证和加密等安全服务。安全子系统可以支持一个或者多个安全模型,目前只支持基于用户的安全模型 USM。
- 访问控制子系统(access control subsystem):通过访问控制模型提供对本地文件访问的授权服务。目前支持基于视图的访问控制模型(view-based access control model,VACM)。

SNMP 利用 SNMP 引擎提供的服务来执行 SNMP 实体的操作,这些应用包括如下。

- 命令生成器(command generator):根据不同的应用生成不同的 SNMP 命令,如 SNMP GET、SNMP GETNEXT 等。

- 命令应答器(command responder):在收到不同的 SNMP 命令后,通过应用程序返回需要的管理数据。
- 通知产生器(notification originator):生成 Trap 或 Inform 等命令。
- 通知接收器(notification receiver):接收并处理 Trap 或者 Inform 命令。
- 代理转发器(proxy forwarder):在不同的 SNMP 实体之间转发消息。

SNMPv3 架构的一个突出特点是采用了可扩充的模块设计的思想,这些模块之间相互协作,根据实现功能的不同,采用相应的模块。SNMPv3 体系结构中单独的模块可以根据实际情况升级而不影响其他模块。

2. 数据封装

如图 7.28 所示,SNMPv3 在原 SNMPv1 或 SNMPv2 的协议数据单元(protocol data unit,PDU)中增加了用于安全保护的报头,利用该报头对原始 SNMP 消息(即 SNMPv1 或 SNMPv2 的 PDU)进行加密、鉴别(具体方式可选)处理,形成 SNMP v3 报文。SNMP v3 报文封装在 UDP 协议中进行传输。SNMPv3 报文由版本信息、头部数据、消息安全参数及 scopedpdu 这 4 部分构成。

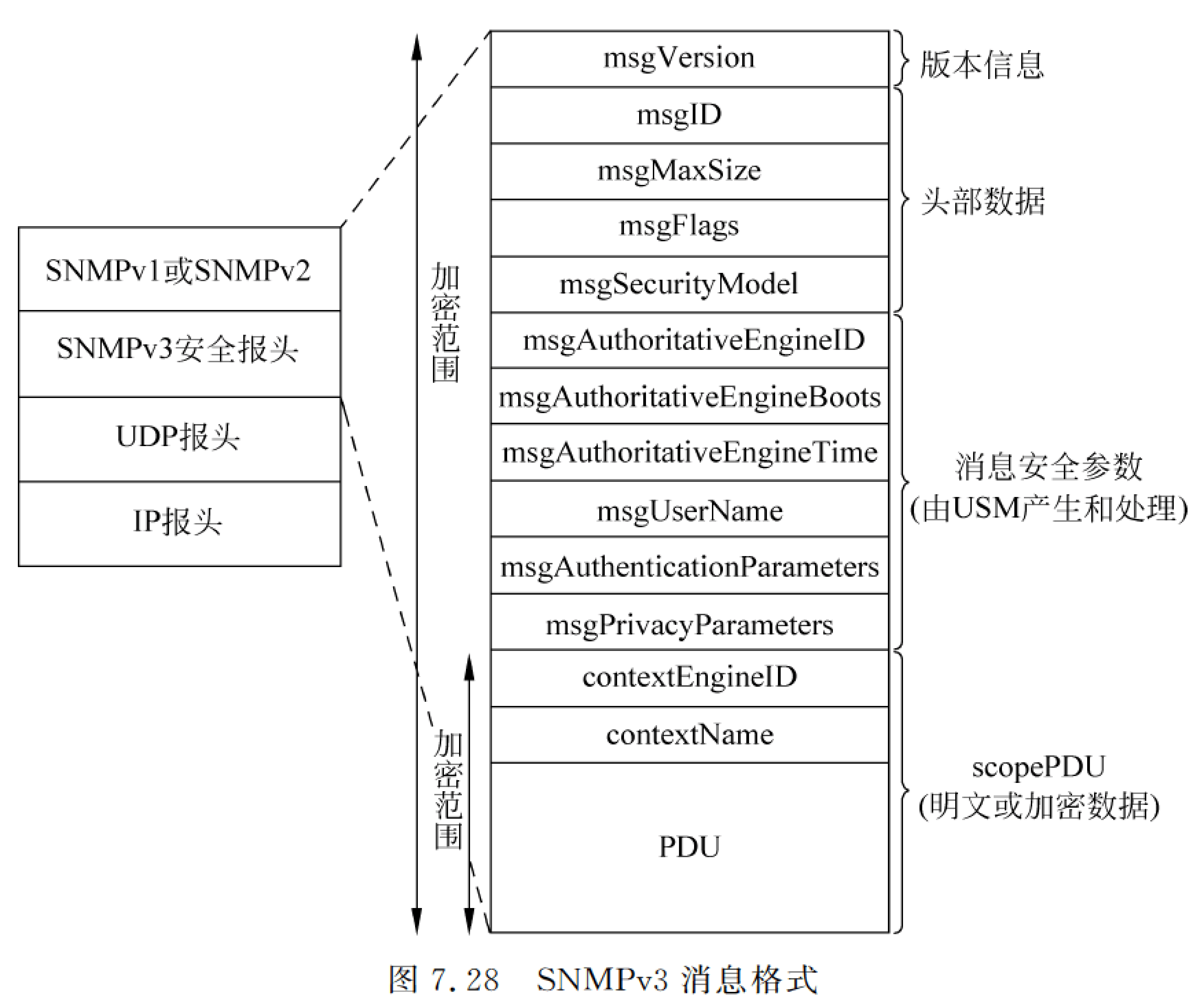


图 7.28 SNMPv3 消息格式

(1) 版本信息

版本信息通过 msgVersion 字段标识。设置为 snmpv3,表示该消息是一个 SNMPv3 消息。

(2) 头部数据

头部数据描述消息标识符 ID、消息最大长度、消息标志和消息安全模型等信息，包括 msgID、msgMaxSize、msgFlags 和 msgSecurityModel 字段。

- msgID：用于通信的两个 SNMP 实体唯一标识一对请求与响应消息，该请求消息与响应消息拥有相同的 msgID 值。
- msgMaxSize：消息发送者能支持的消息的最大字节数，也是该实体能接收的最大字节数。其取值范围为 484～(2^31-1)字节。
- msgFlags：一个 8 位组，目前包含了 reportableFlag、authFlag 和 privFlag 三个标志位，用于控制消息的处理行为。authFlag 和 privFlag 合起来就定义了安全级别(securityLevel)。目前有不认证不加密(noAuthNoPriv)、认证不加密(authNoPriv)和既认证又加密(authPriv)三种安全级别，其安全级别依次提高。
- msgSecurityModel：标识发送端使用的安全模式，接收端只有使用同样的安全模式才能处理该消息。SNMP 的不同版本有不同的安全模式，目前主要有 SNMPv1(1)、SNMPv2c(2)和 USM(3)。对于 SNMPv3 消息，其安全模式为 USM，即基于用户的安全模式。

(3) 消息安全参数

消息安全参数描述用户名及与授权引擎、认证和加密有关的安全参数，是为了支持安全机制而设置的，包括 msgAuthoritativeEngineID、msgAuthoritativeEngineBoots、msgAuthoritativeEngineTime、msgUserName、msgAuthenticationParameters 和 msgPrivacy Parameters 字段。

- msgAuthoritativeEngineID：表示参与消息交互的权威引擎的 snmpEngineID 值。权威引擎是指不需要响应的消息(如 Trap、Response)的发送引擎，或是需要响应的消息(如 Get、GetNext、GetBulk、Set 和 Inform)的接收引擎。
- msgAuthoritativeEngineBoots：表示参与消息交互的权威引擎的 snmpEngineBoots 值，即权威引擎自从配置了 snmpEngineID 以后，重新启动的次数。
- msgAuthoritativeEngineTime：表示参与消息交互的权威引擎的 snmpEngineTime 值，即权威引擎最近一次重启到现在所经过的时间秒数。当其达到最大值(2^31-1)时，snmpEngineBoots 的值加 1，snmpEngineTime 的值被置为 0 并重新开始计数。
- msgUserName：用户名，用于报文的身份认证。
- msgAuthenticationParameters：进行消息认证的消息认证码，长度为 12 个 8 位组。如果不需要认证，则其值为空。
- msgPrivacyParameters：加密参数，用于生成进行 CBS-DES 加解密的初始化矢量(initialization vector，IV)。如果不需要加密，则其值为空。

(4) scopedpdu

scopedpdu 描述上下文信息和协议数据单元，是受加密保护的对象，包括 contextEngineID、

contextName 和 PDU 字段。

- contextEngineID：用于在一个管理域中标识 SNMP context 所处的 SNMP 实体。SNMP context，即 SNMP 上下文，是能被一个 SNMP 实体访问的管理信息的集合。该集合可以跨越多个 SNMP 实体，由多个 SNMP 实体上的可访问管理信息组成。
- contextName：用于在一个 SNMP 实体内部命名一个上下文，在同一 SNMP 实体内 contextName 必须是唯一的。
- PDU：即 SNMPv1 或 SNMPv2 的协议数据单元。

7.5.3 SNMPv3 安全服务的实现

为实现提高网络管理协议安全性的目标，SNMPv3 提出了基于用户的安全模式 USM。USM 提供了身份认证、消息加密和时间窗校验等安全服务。与此同时，SNMPv3 将安全与管理相结合，提出了基于视图的访问控制模式 VACM 来提高 SNMPv3 系统的用户管理能力。VACM 可以在 PDU 级提供安全服务，负责控制用户可以访问的被管对象，以及可以进行的访问操作。

1. USM

在 SNMPv3 的安全通信中，USM 提供了身份认证、消息加密和时间窗校验等安全服务。USM 使用消息认证码 MAC 对交互的消息进行消息源鉴别和完整性鉴别，以防止用户假冒和篡改消息。可以使用的认证协议有 HMAC-MD5、HMAC-SHA1。同时，USM 使用加密方法来保证消息在传输中的机密性，IETF 建议使用 CBC-DES 加密协议。此外，USM 使用时限检查机制来防止消息滞延和重放。USM 为 SNMP 实体维护了一个用户信息表 usmUserTable，包含了用户的相关属性信息，如 UserEngineID（与用户通信的权威引擎的 SnmpEngineID）、userName（用户名）、securityName（安全名）、authProtocol 和 authKey（用户使用的认证协议和密钥）、privProtocol 和 privKey（用户使用的加/解密协议和密钥）等。通信双方实体通过所维护的用户信息，对收发的消息进行安全处理。

此外，在使用认证与加密服务时，通信的 Manager 和 Agent 须共享用户的认证密钥和加密密钥。然而，一个用户可能拥有多个 Agent 的访问和控制权限，如果该用户在每个 Agent 上的密钥都相同，必将导致安全隐患。但是让用户记忆大量的密钥，又会增加用户的负担。为此，USM 采用了密钥本地化机制，用户只需记忆自己的认证密码和加密密码，由 USM 负责将密码转换成认证协议和加密协议所需的本地化的密钥。采取这种方式，可以缓解字典式密钥攻击的速度。同时，由于不同用户的密钥各不相同，并且同一个用户在不同 Agent 上的密钥也不同，因此一个用户在某一个 Agent 上的密钥受到损害不会影响到其他的 Agent。

下面具体说明 USM 中各项安全服务的实现方式。

(1) 身份认证和数据完整性保护

SNMPv3 中,身份认证通过数字签名实现,提供消息完整性和数据源鉴别。USM 使用简单的 HMAC 算法进行消息的签名(类似于 TSIG 中的签名机制,不同于利用 PKI 的签名机制)。

身份认证的实现方法是:管理者与代理共享一个对称密钥,此密钥的密码通过 MD5 或 SHA 算法获得。管理者以该密钥和待传输的消息作为 HMAC 算法的输入,通过哈希算法计算出 MAC 值,然后将其加入消息的认证参数字段(msgAuthenticationParameters)中传输。代理收到该消息后使用同一密钥计算该消息的 MAC 值,并与收到的 MAC 进行比较。如果相同则认为消息可靠,否则拒绝接收该消息并返回错误信息。

(2) 合时性检查

合时性的意思是消息一旦被认定是鉴别过的,就必须及时接收并处理,从而防止消息被延迟或重复。

合时性通过同步时钟来保证。在两个引擎之间每一次安全的 SNMPv3 通信中,一个引擎被看作是命令式的,另一个则是非命令式的。命令式引擎维持一个"时钟"值用于同步,而非命令式引擎的任务就是获取并跟踪这个"时钟"值。"时钟"值由 EngineBoots 和 EngineTime 两部分组成,其中 EngineBoots 表示引擎重新启动的次数,EngineTime 表示引擎最后一次重新启动后所经历的秒数。

合时性检查的实现方法是:初始时,非命令式引擎发送一个请求消息给命令式引擎,将其中 msgAuthoritativeEngineBoots 和 msgAuthoritativeEngineTime 都置为 0,从而获取命令式引擎的 EngineBoots 和 EngineTime 的值。命令式引擎的响应是发送一个报告消息,其中包含最新的 SnmpEngineBoots 和 SnmpEngineTime 值。非命令式引擎一旦获取了命令式引擎的 SnmpEngineBoots 和 SnmpEngineTime 值,就需要跟踪这些值。以后发送给命令式引擎的消息将包含有这样的值:非命令式引擎"认为"命令式引擎的 SnmpEngineBoots 和 SnmpEngineTime 值应该是多少。命令式引擎则将这样的值与自己的实际值比较。如果比较结果相差 150s 以内,消息被认为是及时的;如果相差超过 150s,该消息将被丢弃。

(3) 重复性检查

SNMPv3 报文头部有一个报文标识符 msgID,发送 SNMP 请求的实体将负责使用这个消息标识符,将其接收到的一个响应与一个未解决的请求进行匹配。重复性的意思是指在 150s 间隔中不能有两个相同的 msgID 字段的报文回送。

假设在一个 150s 时间窗口中接收了两个具有相同标识符的响应,第一个响应将被匹配,然后删去具有该标识符的请求。因此,后到的重复响应将因为无响应请求与之匹配而被丢弃。通过这种机制,发送者可以在一定程度上判别报文是否是重放的,以防止报文复制和重放攻击。

(4) 机密性

与身份认证类似,机密性保护也需要管理站和代理共享同一个对称密钥(称为

privKey)，以实现消息的加密和解密，该对称密钥由用户密码经 MD5 算法获得。

USM 加密使用的是 DES 的 CBC 模式，用 DES CBC 算法加密和解密需要 DES 密钥和一个初始向量 IV。DES 对称密钥由 privKey 的前 8 个字节形成(由于 DES 只需要 56 位的密钥，因此每个字节的最低有效位被忽略掉)，后 8 个字节则用作 pre-IV。IV 由 pre-IV 和一个 salt 值按位进行“异或”而产生。其中 salt 值是由该引擎 SnmpEngineBoots 的当前取值与本地加密协议维护的一个整数串接而成。为了防止 IV 泄露，将 salt 值放在消息的 msgPrivacyParameter 字段中传输，接收方根据该字段计算出正确的 IV，再进一步对密文进行解密。

2. VACM

VACM 在 SNMPv3 安全通信中提供 PDU 级别的安全服务，负责控制用户可以访问的被管对象，以及可以进行的访问操作。为提供访问控制服务，VACM 需定义组(groups)、安全级别(securityLevel)、上下文(contexts)、MIB 视图(MIBViews)及访问策略(access policy)5 个基本要素。

- 组：包含一个或多个由<安全模型，安全名>二元组标识的用户，同组的所有用户具有相同的访问权限，但每个用户只能加入一个组。每个组由组名(groupName)标识。
- 安全级别：表示在进行访问权限控制时所需使用的安全级别，共有不认证不加密(noAuthNoPriv)、认证但不加密(authNoPriv)及既认证又加密(authPriv)三种。同组的用户可以定义不同的安全级别。
- 上下文：是能被一个 SNMP 实体访问的管理信息的集合，该集合可以跨越多个 SNMP 实体，由多个 SNMP 实体上的可访问管理信息组成。同时，同一管理信息可以属于多个上下文。每个上下文由上下文名(contextName)标识。
- MIB 视图：是一个管理对象的集合，由一个或多个视图子树组成。而一个视图子树是一系列具有相同 OID 前缀的 MIB 对象实例。
- 访问策略：用于定义一个组的访问权限。在一个某组通过使用某特定的安全模型和安全级别可以访问的上下文中，通过使用一个读视图(read-view)、写视图(write-view)和通告视图(notify-view)来定义该组在这个上下文中的访问权限。其中，读视图表示该组能进行读操作的 MIB 视图；写视图表示该组能进行写操作的 MIB 视图；通告视图表示该组能进行通告的 MIB 视图。

此外，为实现访问请求控制功能，VACM 还需使用 4 张表：上下文表(vacmContextTable)、组映射表(vacmSecurityToGroupTable)、访问控制表(vacmAccessTable)和视图子树表(vacmViewTreeFamilyTable)来实现访问控制机制。上下文表用于存储本地所有可被访问的上下文的名字；组映射表用于存储<安全模型，安全名>二元组与组名的对应关系；访问控制表用于存储各组的访问权限，指定了某个组在特定的上下文中，使用特定的安全模型

与安全级别,能访问哪些读视图、写视图及通告视图;视图子树表用于存储 MIB 视图内包含的哪些视图子树,以及不包含哪些视图子树的相关信息。

本章实验

1. 利用 PGP 保护电子邮件的安全。
2. 利用 BIND 配置实现 DNSSEC 和 TSIG。

思考题

1. DNSSEC 和 TSIG 有什么区别和联系?
2. 同样是保护电子邮件的安全协议,从协议层次和封装顺序上看,PGP 和 S/MIME 的区别是什么?
3. PGP 和 S/MIME 的证书格式有什么不同?
4. PGP 和 DNSSEC 的信任关系模型有什么不同?
5. PGP 的公钥环和私钥环文件的结构是怎样的? 在 PGP 中为什么要使用多个私钥和公钥对? 这一点和 PKI 有什么区别?
6. 假设你为 WWW 服务器设计一个安全的解决方案,那么应该从哪些方面考虑提高其安全性的问题?

第三部分

网络安全技术与应用

第8章

企业级安全技术

8.1 虚拟专用网

8.1.1 VPN概述

RFC 2764 对虚拟专用网(VPN)的定义是：利用公用网络(通常是 Internet)将异地的站点或用户互连而形成的一个具有私有(专用)性的网络。这种私有性是指 VPN 可以保证其上通信的私有数据不被未授权访问，这可以通过认证、加密或路由隔离等机制实现。为了提供这种私有服务，VPN 网络依赖 VPN 协议实现，例如可以使用 IPSec 或 PPTP(point-to-point tunneling protocol)等协议实现数据加密服务和身份鉴别机制。

如图 8.1 所示，VPN 使用公共网络基础设施传输私有数据，而不使用专用的私有线路(例如使用图 8.2 所示的专用线路)。也就是说，VPN 没有自己的专用链路和网络基础设施，但通过 VPN 协议它可提供与专用网络相同的安全服务，因此称为虚拟专用网。

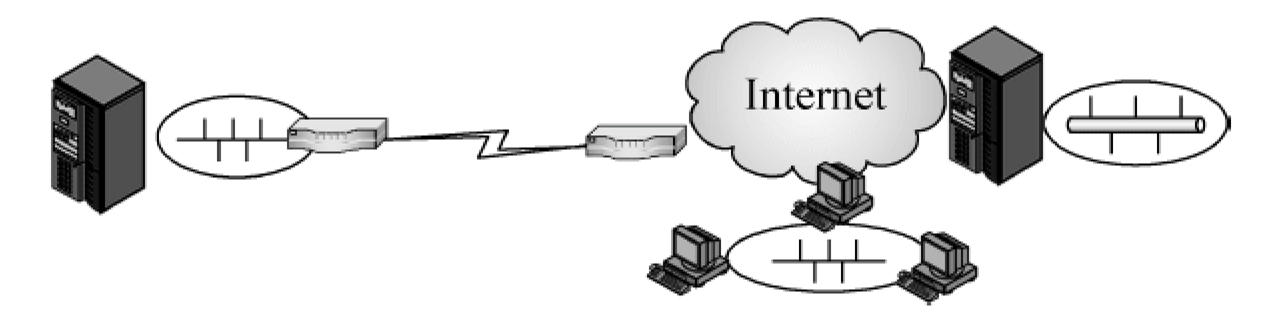


图 8.1　利用 Internet 的 VPN 网络

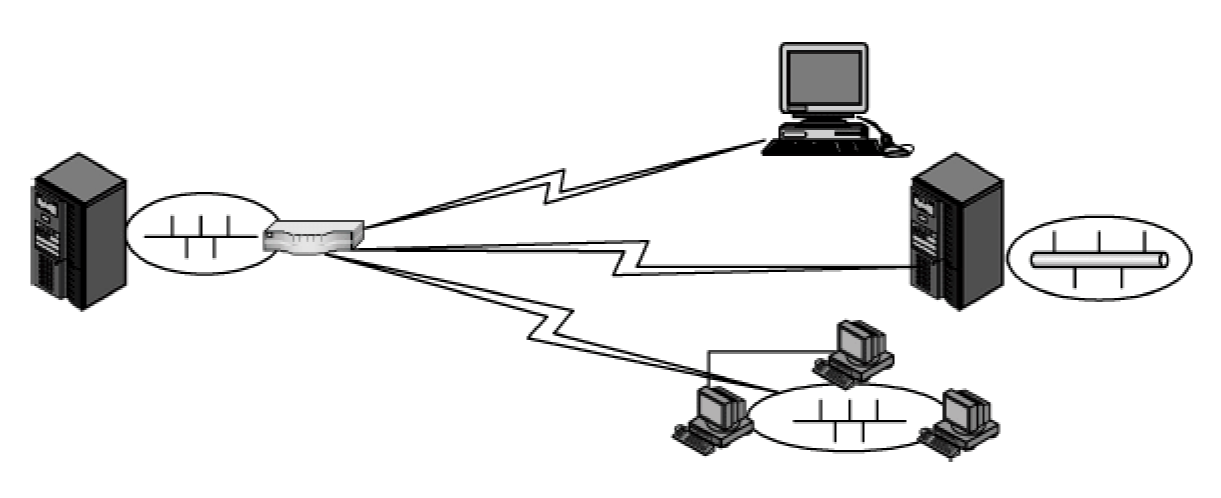


图 8.2　专用网络

VPN 可提供如下安全服务中的一种或多种。

- 访问控制
- 认证
- 机密性
- 数据完整性

实现 VPN 的典型技术有两种：隧道技术和虚拟路由技术。采用隧道技术实现 VPN 的基础框架如图 8.3 所示。在 VPN 设备(VPN device)之间建立 VPN 隧道(VPN tunnel)，VPN 设备是实现 VPN 协议的对等实体，可以使用路由器、防火墙或 RAS(remote access server)服务器实现。VPN 隧道实际上是采用某种协议(即隧道协议)对网络数据包进行封装，从而提供安全特性。VPN 隧道可以使用多种协议实现，例如可以使用虚拟的 PPP 连接在公共网络上传输由隧道协议封装的用户数据，即 PPTP 协议。

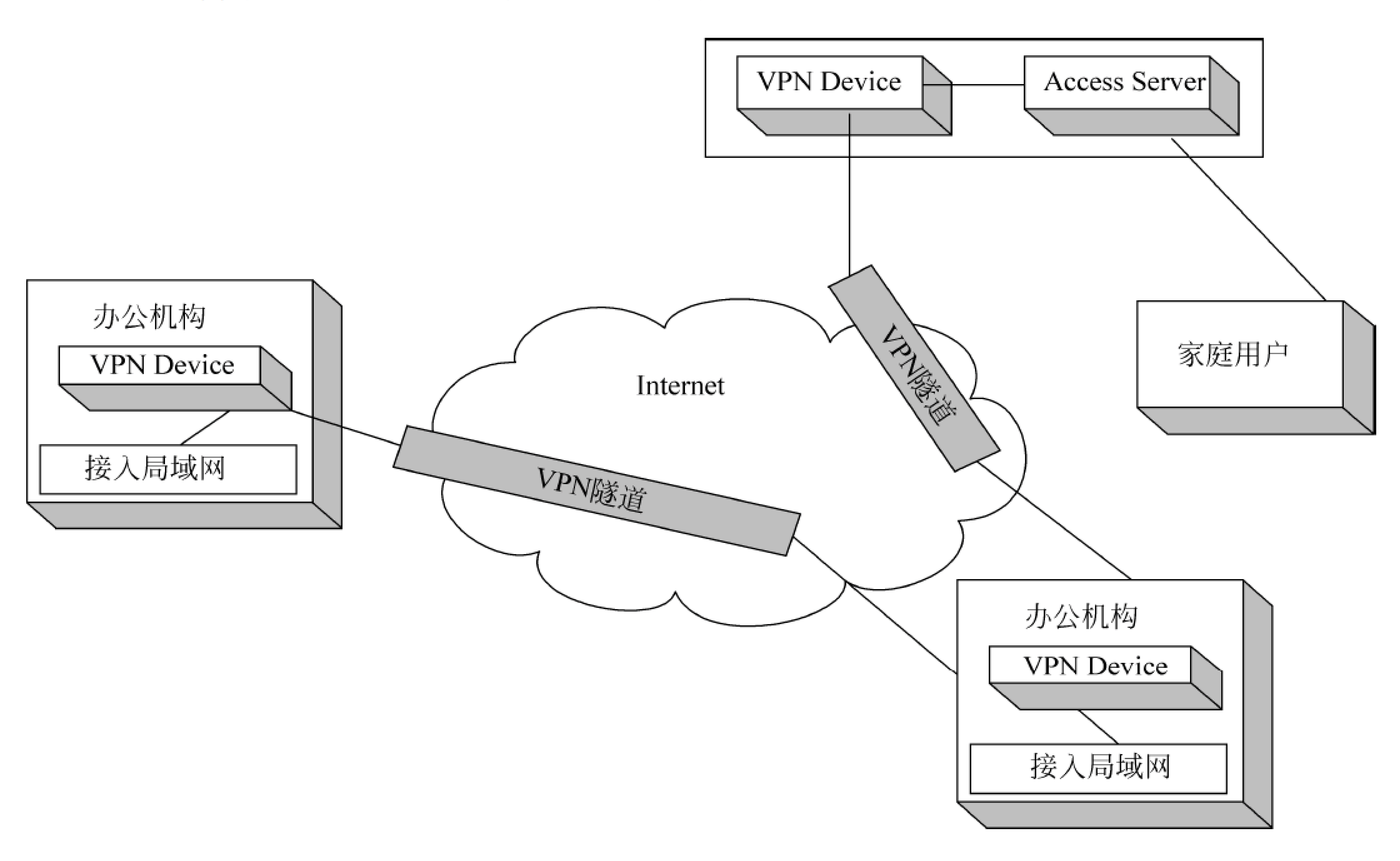


图 8.3　隧道技术的 VPN 基础构架

图 8.4 给出了使用隧道实现 VPN 的网络拓扑，其中公司总部和 VPN 客户端(可以是移动用户、拨号用户或局域网用户)位于异地。公司总部的防火墙提供 VPN 服务，而 VPN 客户端使用 VPN 客户端软件接入 VPN 服务器，接入后即产生 VPN 隧道，远程的公司用户可以通过隧道协议安全地访问公司总部信息。

隧道技术将用户数据包采用隧道协议进行重新封装，并在公用网络中传输私有 VPN 数据。封装后的数据仍采用公共网络使用的协议(例如 IP 协议)进行传输，传输过程对公共网络节点(如路由器)透明，即公用网络中的路由节点不会知道所传输的数据是否是用于专

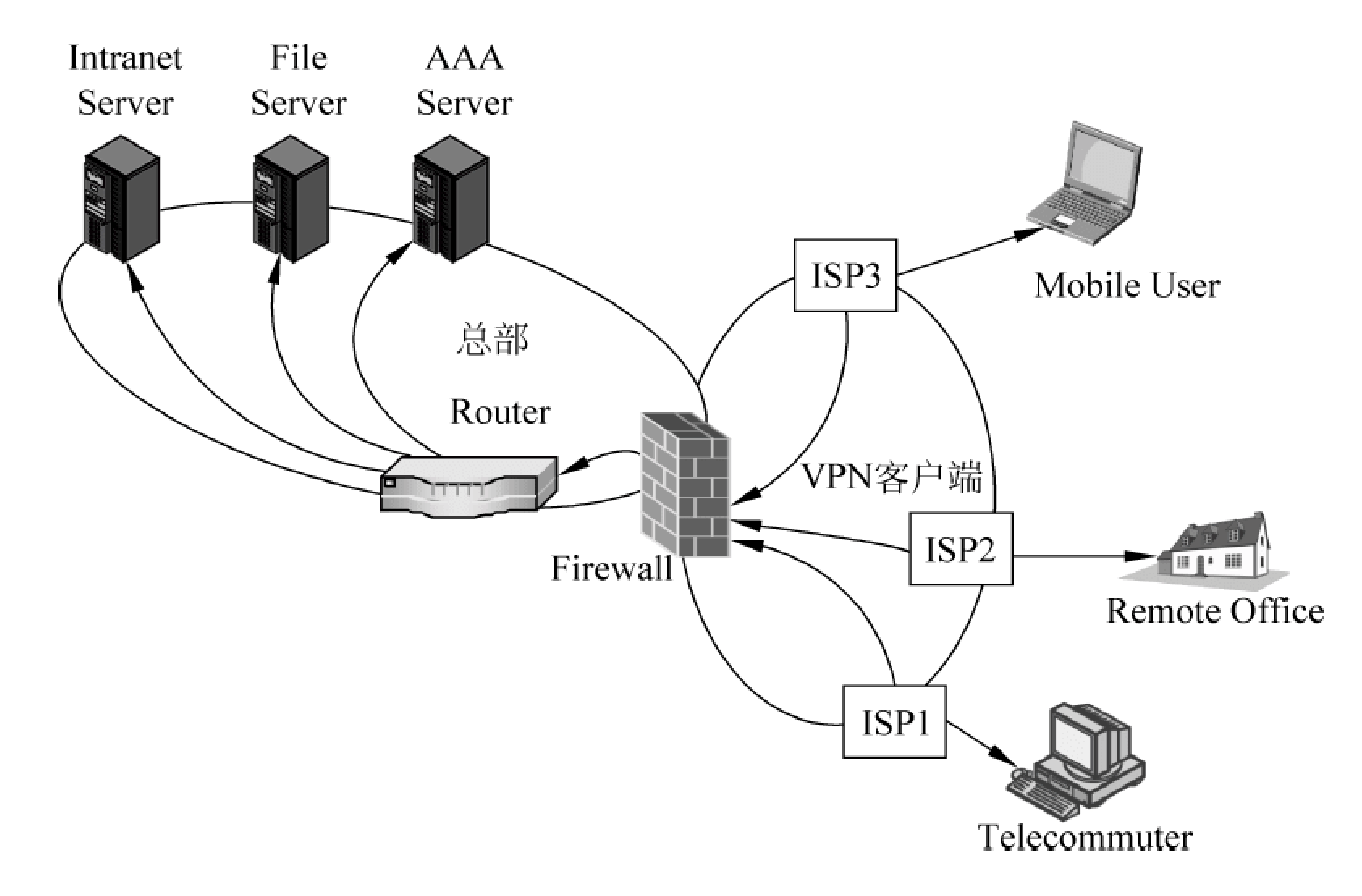


图 8.4 隧道技术的 VPN 网络拓扑

用网络的。

另一种实现 VPN 的技术是虚拟路由(virtual route,VR)或虚拟路由器技术。虚拟路由器在软件层面上效仿物理路由器。虚拟路由器有其各自的 IP 地址和转发表,并且它们的路由是相互独立和隔离的。从用户的角度出发,虚拟路由器的功能和物理路由器是相同的。虚拟路由器正是利用路由信息隔离的特性为 VPN 用户提供数据私有性服务的。

如图 8.5 所示,VPN-1 连接在虚拟路由器 VR-1 中,而 VPN-2 连接在虚拟路由器 VR-2 上,VR-1 和 VR-2 的路由表是各自独立和相互隔离的。因此,VPN-1 和 VPN-2 之间相当于处在两个不同子网中。而 VR-1 的本地和远程网络之间可以通过因特网交换路由信息,相当于处在同一个子网中,可以互相访问。而无论是本地还是远程的 VR-2 连接的网络用户都不能访问 VR-1 网络的数据,因为他们没有该网络的路由信息。

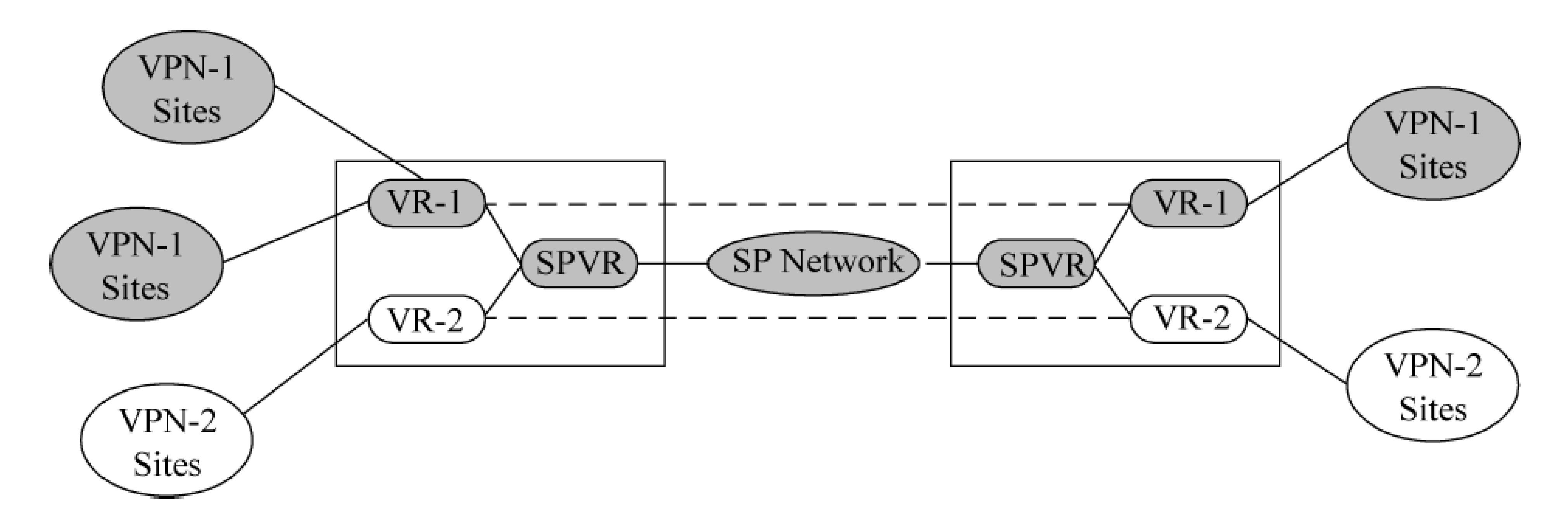


图 8.5 基于虚拟路由器的 VPN

8.1.2 VPN 分类

有多种 VPN 的分类方法，如按网络拓扑、按实现 VPN 的协议层次和按用户特征划分等。

(1) 按网络拓扑划分

按照实现 VPN 的网络拓扑不同，可以将 VPN 网络划分为远程访问(remote access)和工作地到工作地(site to site)两种。

① 远程访问。

如图 8.6 所示，远程终端用户通过 VPN 连接到总部网络中，并和总部网络进行安全通信。例如，一个公司雇员从家中利用拨号接入因特网，再拨入公司总部的 VPN 服务器，则他可以安全地从远程访问公司的机密资料。

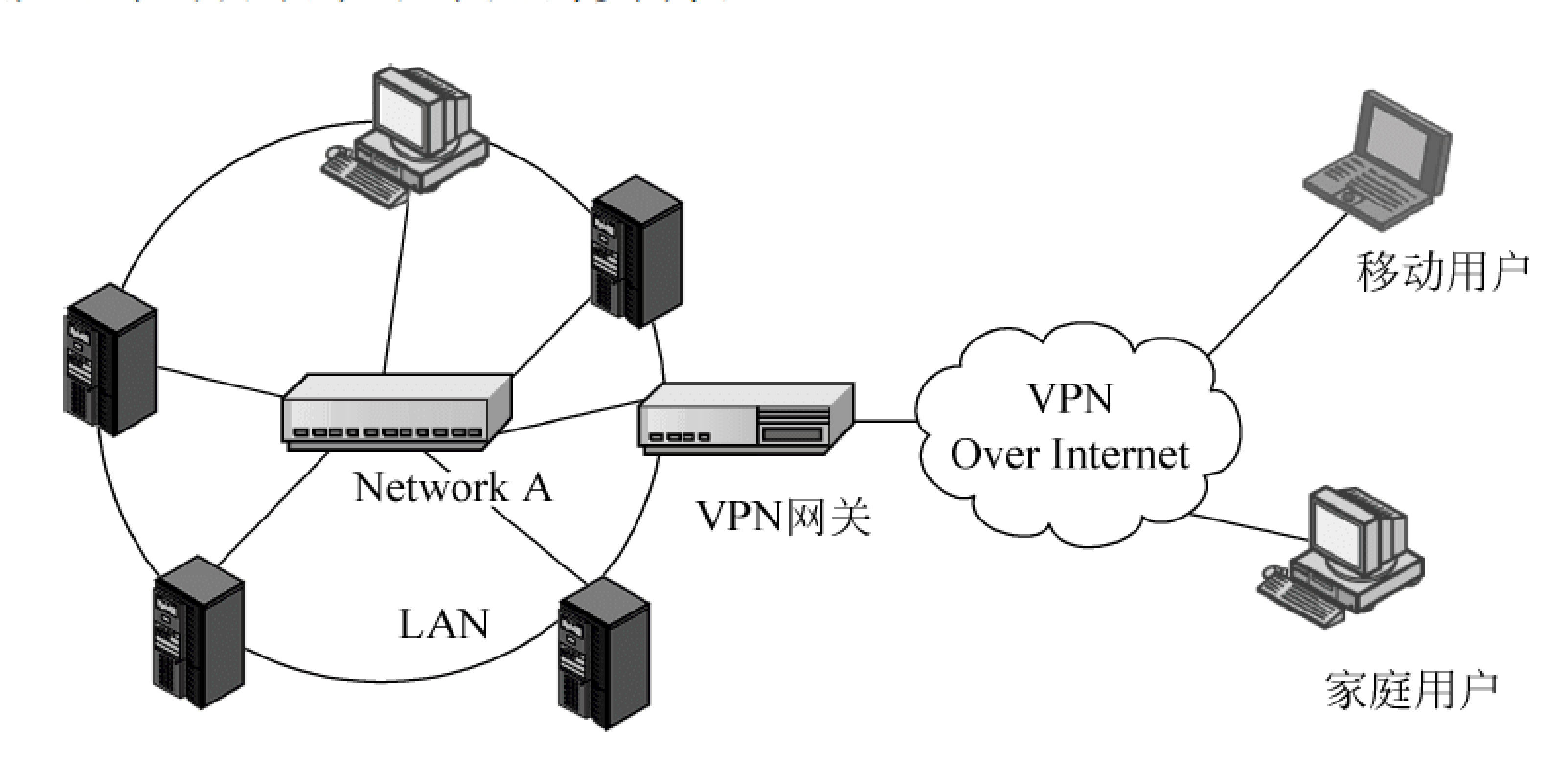


图 8.6　远程访问 VPN

这种网络拓扑中，如果采用隧道技术，VPN 隧道将在终端用户(VPN 客户端)和总部网络的边缘设备(即 VPN 服务器，如路由器、防火墙和 NAS 等)上建立。客户端使用 VPN 客户端软件拨入 VPN 服务器。

② 工作地到工作地。

如图 8.7 所示，两个局域网通过 VPN 互连。例如，一个公司具有两个地理位置不同的办公地点，这两个办公地点通过 VPN 互连，通过因特网进行安全通信。如果采用隧道技术，VPN 隧道将在两个分部门网络的边缘设备(如路由器、防火墙和 NAS 等)上建立。

(2) 按协议层次划分

理论上，VPN 可以在 Internet 的任何协议层(TCP/IP 参考模型)实现，包括 2 层 VPN(如 PPTP、L2F 和 L2TP)、2.5 层 VPN(如 MPLS VPN)、3 层 VPN(如 IPSec VPN)、4 层 VPN(如位于应用层和传输层之间的 SSL/TLS、SSH 和 SOCKS)及 5 层 VPN(如应用层代理、DNSSec 等)。如图 8.8 所示。

(3) 按用户划分

① 企业内联网(Intranet)。

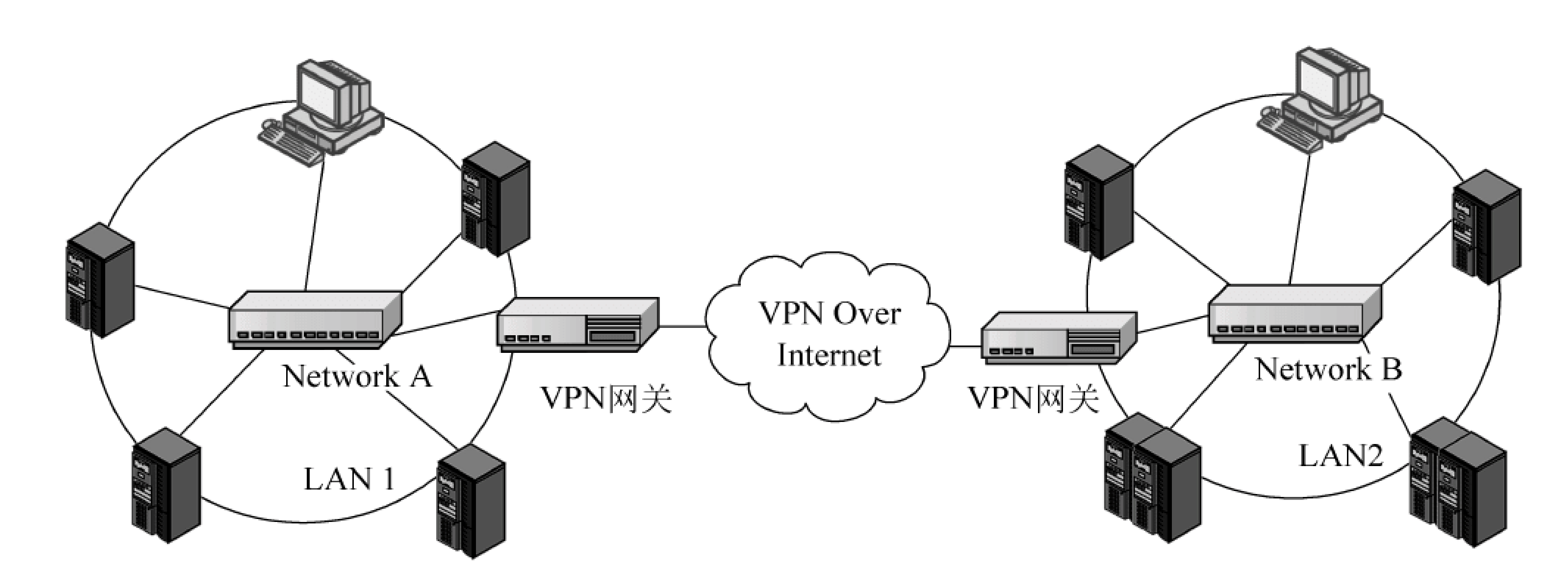


图 8.7　工作地到工作地 VPN

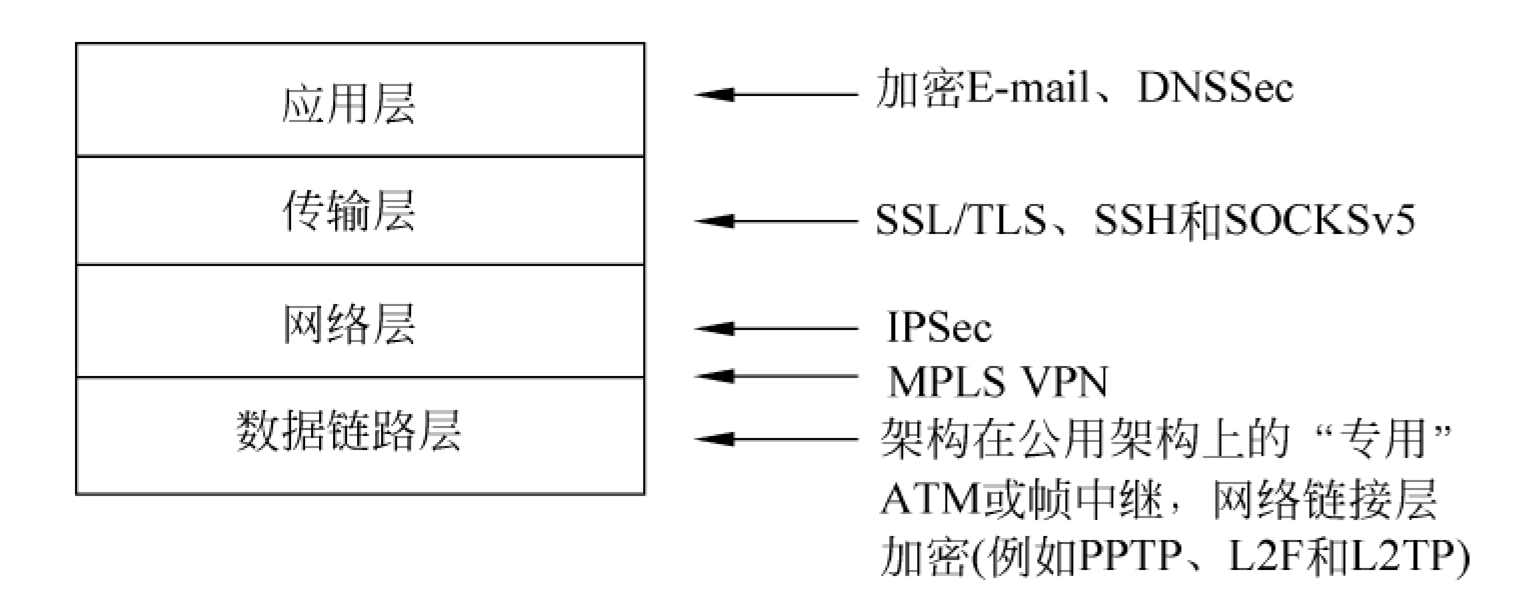


图 8.8　VPN 在各层的实现

这种网络中只允许组织内部的用户之间通过 VPN 进行通信。可以将一个内部网络地址分配给远程的工作场地或是计算机，这样拥有这个专用网络地址的主机将被当作本地用户对待，可以在局域网中共享数据。

② 企业外联网(Extranet)。

本地和远程 VPN 用户之间有限制地相互访问对方的资源。这种网络中需要区别远程 VPN 用户和本地用户，通过访问控制、身份认证和授权机制实现他们对不同资源的不同访问权限。

VPN 可采用不同的协议实现，如 IPSec、PPTP、L2F/L2TP 及 MPLS 等。IPSec 隧道协议见第 5 章，下面介绍 PPTP、L2F、L2TP 及 MPLS VPN 技术。

8.1.3　PPTP

点对点隧道协议(PPTP)由 RFC 2637 定义，它利用二层协议 PPP 承载 VPN 业务。PPTP 最早由 Microsoft 提出，得到 Ascend、3COM 等公司支持，如 Windows NT 4.0 以上版本的操作系统中都提供了该协议的实现，包括 PPTP VPN 客户端和服务器。PPP 支持多种网络协议，可把 IP、IPX、AppleTalk 或 NetBEUI 的数据封装在 PPP 报文中，再将 PPP

报文封装在 PPTP 隧道协议中，最后在 IP 网络中进行传输。

PPTP 通过 PPTP 控制连接来创建、维护和终止一条隧道，并使用通用路由封装(generic routing encapsulation，GRE)对 PPP 帧进行封装，GRE 即 PPTP 隧道协议。封装前，PPP 帧的有效载荷即有效传输数据首先必须经过加密、压缩或是两者的混合处理。

PPTP 使用的认证机制与创建 PPP 连接时相同，包括 PAP、EAP、CHAP 、MS-CHAP 和 Shiva 密码字认证协议(shiva password authentication protocol，SPAP)等。PPTP 继承 PPP 有效载荷的加密和压缩方法。在 Windows 操作系统中，由于 PPP 帧使用微软点对点加密技术(Microsoft point-to-point encryption，MPPE)进行加密，因此认证机制必须采用 EAP 或 MS-CHAP。MPPE 只提供连接加密，而不提供端到端的加密。如果应用中要求实现端到端加密，则可在 PPTP 隧道建立之后，使用 IPSec 对两端的 IP 数据流进行加密处理。

1. PPP 工作过程

如图 8.9 所示，PPTP 的工作过程描述如下。

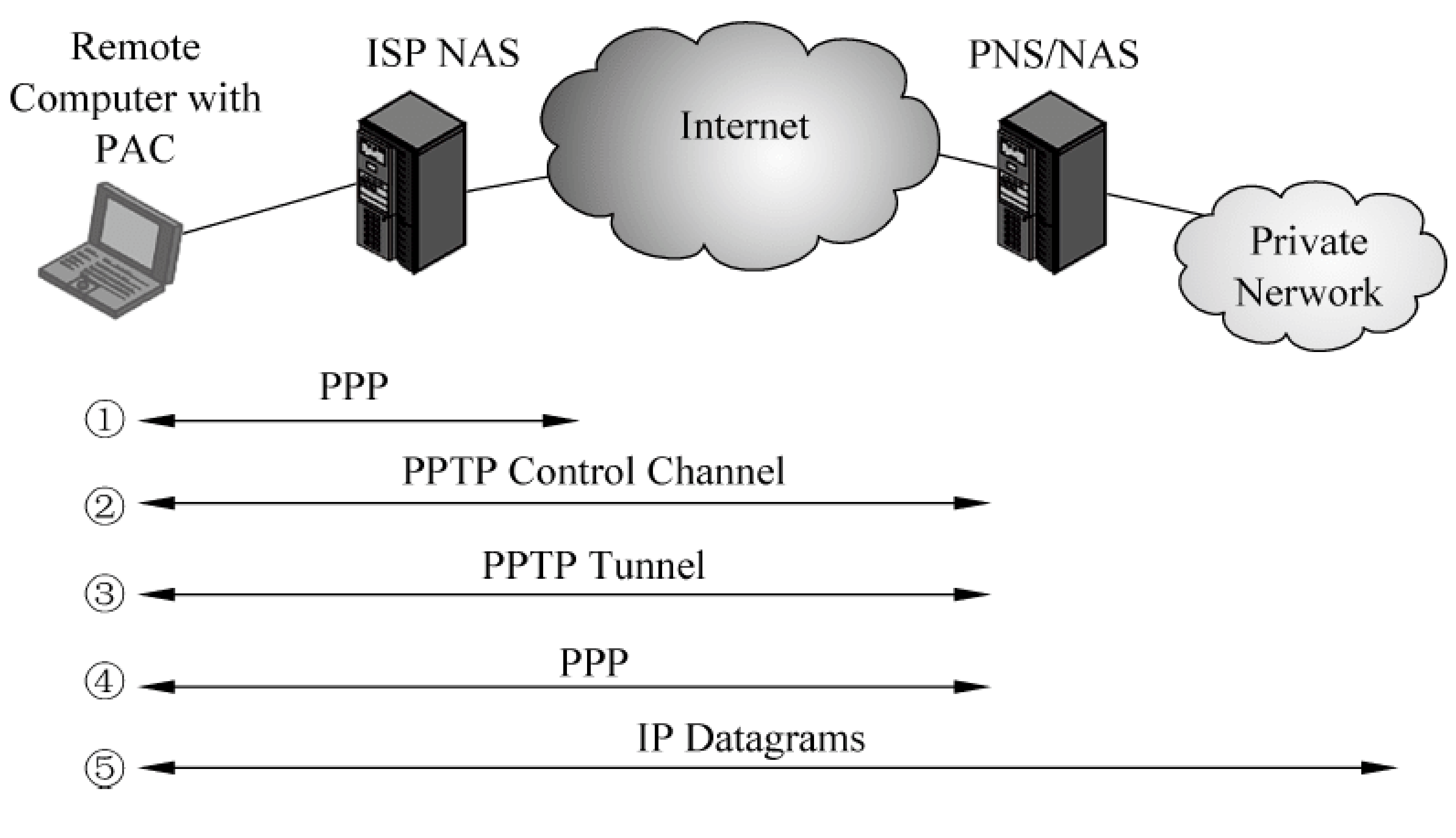


图 8.9 PPTP 工作过程

图 8.9 中，PPTP VPN 客户端称为 PAC(PPTP access concentrator)，由用户的操作系统集成(如 Windows 的 VPN 拨号网络客户端)。PPTP 服务器称为 PNC(PPTP network server)，图中为内部网络的 NAS 集成。ISP NAS 提供用户拨号访问互联网服务。

① 远程计算机通过 PPP 拨号连接到本地 IPS 网络接入服务器 NAS 上(这一步并不是必需的，用户也可以通过 LAN 等其他的方式接入 Internet)。

② PPTP 的客户端 PAC 通过 Internet 建立和 PPTP 服务器 PNS 之间的控制连接(通过 TCP 协议)。即 VPN 客户端通过因特网和 VPN 服务器建立 TCP 连接，然后传输 PPTP 控制信息。

③ PPTP 隧道的通信参数通过这个控制连接进行协商，协商完毕后 PPTP 隧道被建立。

④ 远程计算机通过刚才建立的 PPTP 隧道建立第二个经由 PAC 到 PNS 的 PPP 连接，

从而接入到专用网络的 NAS。这时，PNS 可以为远程客户端分配内部网络 IP 地址，并且可以通过 CHAP 等机制来鉴别客户端或服务器。

⑤ 由 GRE 封装的 PPP 帧通过 IP 协议在 Internet 中传输。

2. PPTP 控制连接与隧道维护

PPTP 的控制信息经过 PPTP 控制连接传输，PPTP 控制连接使用 TCP 协议在 PPTP 客户端和(PAC)PPTP 服务器(PNS)之间建立，PPTP 客户机使用动态分配的 TCP 端口号，PPTP 服务器则使用保留 TCP 端口号 1723。控制连接的初期将建立 PPTP 控制连接自身，后期负责建立 PPTP 隧道(称为 PPTP 呼叫)。

PPTP 控制信息包括 PPTP 控制连接管理和 PPTP 呼叫管理两种。PPTP 控制连接管理负责 PPTP 控制连接自身的建立、维护和终止，包括周期性地发送回送请求和回送应答消息，以检测客户机与服务器之间可能出现的连接中断。PPTP 呼叫管理负责 PPTP 隧道的建立、维护和终止。

如图 8.10 所示，PPTP 控制消息包括一个 IP 报头，一个 TCP 报头和 PPTP 控制信息。图中所示的 PPTP 控制连接数据包还包括数据链路层报头和报尾，该数据链路层由公共基础设施如因特网的通信子网提供。

数据链路层报头	IP	TCP	PPTP 控制信息	数据链路报尾

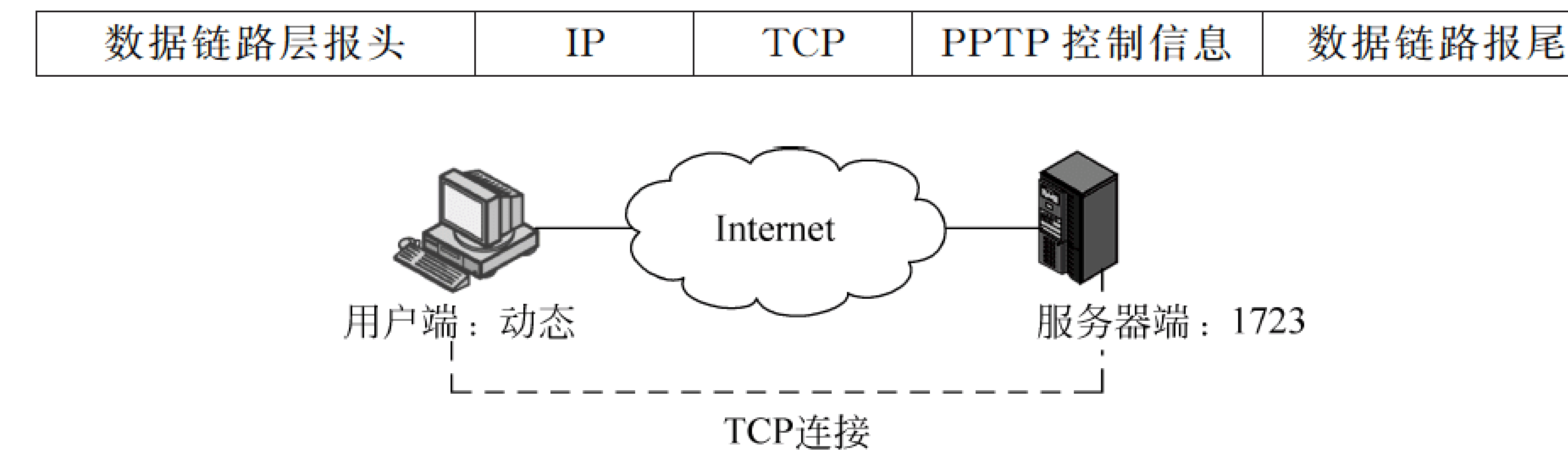


图 8.10　PPTP 控制消息的封装

PPTP 控制信息的数据包格式如图 8.11 所示。

16 位	32 位
长度	PPTP 消息类型
Cookie	
控制信息类型	保留 0
协议版本	保留 1
帧类型	
承载性能标识	
最大会话数	固件修订版本
DNS 名称	
供应商字串	

图 8.11　PPTP 控制消息格式

- 长度：该 PPTP 信息的 8 位总长，包括整个 PPTP 头。
- PPTP 消息类型：可能值如下。
  —— 控制信息
  —— 管理信息
- Cookie：以连续的 0x1A2B3C4D 进行发送，其目的是确保接收端与 TCP 数据流间的同步。
- 控制信息类型：包括控制连接管理(Control Connection Management)和呼叫管理(Call Management)两种。其中控制连接管理的可能值如下。
  —— 开始控制连接请求(Start-Control-Connection-Request)
  —— 开始控制连接应答(Start-Control-Connection-Reply)
  —— 停止控制连接请求(Stop-Control-Connection-Request)
  —— 停止控制连接应答(Stop-Control-Connection-Reply)
  —— 回应请求(Echo-Request)
  —— 回应答复(Echo-Reply)
  呼叫管理的可能值如下。
  —— 导出呼叫请求(Outgoing-Call-Request)
  —— 导出呼叫应答(Outgoing-Call-Reply)
  —— 导入呼叫请求(Incoming-Call-Request)
  —— 导入呼叫应答(Incoming-Call-Reply)
  —— 导入—呼叫—连接(Incoming-Call-Connected)
  —— 呼叫清除请求(Call-Clear-Request)
  —— 呼叫—断开连接—通告(Call-Disconnect-Notify)
  —— 广域网错误通告(WAN-Error-Notify)
  —— 设置链路信息(Set-Link-Info)
- 保留 0 & 1：必须设置为 0。
- 协议版本 PPTP 的版本号。
- 帧类型：该信息发送方可以提供。
  —— 异步帧支持(asynchronous framing supported)
  —— 同步帧支持(synchronous framing supported)
- 承载性能标识：该信息发送方可以提供。
  —— 模拟访问支持(analog access supported)
  —— 数字访问支持(digital access supported)
- 最大会话数：该 PAC 可以支持的个人 PPP 会话总数。
- 固件修订版本：若由 PAC 出发，则包括发出 PAC 时的固件修订本编号；若由 PNS 出发，则包括 PNS PPTP 驱动版本。

• DNS 名称：标识 PAC 或 PNS 的 DNS 名称。
• 供应商字串：标识特定供应商字串，指明使用的 PAC 类型或 PNS 软件类型。

以下讲述 PPTP 呼叫与控制连接的创建、维护和终止过程。

(1) PPTP 呼叫与控制连接创建

PPTP 控制连接通过以下步骤建立。

① PPTP 客户机上一个动态分配 TCP 端口与 PPTP 服务器上的 TCP 端口 1723 建立 TCP 连接。

② PPTP 客户端发送一条 PPTP Start-Control-Connection-Request(开始控制连接请求)消息，后者将用于建立一个 PPTP 控制连接。

③ PPTP 服务器使用一条 PPTP Start-Control-Connection-Reply(开始控制连接应答)消息予以响应。

④ PPTP 客户端发送一条 PPTP Outgoing-Call-Request(导出呼叫请求)消息，并选择一个呼叫 ID(call ID)，识别用于将数据从 PPTP 客户端发送到 PPTP 服务器的 PPTP 隧道。PPTP 客户端使用 PPTP Outgoing-Call-Request 消息从 PPTP 服务器请求一个 PPTP 隧道(也称为呼叫)。

⑤ PPTP 服务器发送一条 PPTP Outgoing-Call-Reply(导出呼叫应答)消息，并选择自身的呼叫 ID，识别将数据从 PPTP 服务器发送到 PPTP 客户端的 PPTP 隧道。

⑥ PPTP 客户端发送一条 PPTP Set-Link-Info(设置链路信息)消息来指定 PPTP 协商选项。

PPTP 呼叫与控制连接创建过程的执行结果如下。

• PPTP 服务器已允许创建一个 PPTP 隧道。
• PPTP 客户端已确定在通过 PPTP 隧道向 PPTP 服务器发送数据时在 GRE 报头中使用的呼叫 ID(即隧道标识符)。
• PPTP 服务器已确定在通过 PPTP 隧道向 PPTP 客户端发送数据时在 GRE 报头中使用的呼叫 ID。

(2) PPTP 控制连接维护

为了维持 PPTP 控制连接，不管 PPTP 客户端和服务器之间是否正在发送 GRE 封装的数据，PPTP 客户端每隔 60s 发送一条 PPTP 回应请求(echo request)消息。在收到该请求消息时，PPTP 服务器将发送一条 PPTP 回应答复(echo reply)消息。PPTP 回应请求消息包含一个 Identifier 字段，该字段的值将在 PPTP 回应答复消息中回显，以便 PPTP 客户端能够将 PPTP 回应请求与其应答相匹配。

(3) PPTP 呼叫与控制连接终止

为了终止 PPTP 连接，PPP 连接、PPTP 协议连接(控制连接)和 TCP 连接必须全部终止。下面以 PPTP 客户端发起终止请求为例讲述 PPTP 呼叫与控制连接终止过程。客户端使用如下步骤终止 PPTP 连接。

① PPTP 客户端发送一条 PPTP Set-Link-Info 消息指定链路的 PPP 参数。

② PPTP 客户端发送一条 Link Control Protocol(LCP)Terminate-Request 消息来终止 PPP 连接。LCP 是 PPP 协议族中的一种协议，它负责 PPP 链路连接的配置和维护。

③ PPTP 服务器发送一条 PPTP Set-Link-Info 消息来指定链路的 PPP 参数。

④ PPTP 服务器发送 LCP Terminate-Ack 消息来响应 LCP Terminate-Request 消息，从而终止 PPP 连接。

⑤ PPTP 客户端发送一条 PPTP Clear-Call-Request 消息，向 PPTP 服务器通告 PPTP 控制连接即将终止。

⑥ PPTP 服务器使用一条 PPTP Call-Disconnected-Notify 消息进行响应。

⑦ PPTP 客户端发送一条 PPTP Stop-Control-Connection-Request 消息来终止 PPTP 控制连接。

⑧ PPTP 服务器使用一条 PPTP Stop-Control-Connection-Reply 消息进行响应。

⑨ TCP 连接终止。

3. PPTP 数据封装

PPTP 数据的隧道化过程采用多层封装的方法。图 8.12 显示了在 PPTP 隧道中传输的数据包格式。

帧头	IP 报头	GRE 报头	PPP 报头	加密的 PPP 净载荷	帧尾

图 8.12 在隧道中传输的 PPTP 数据包格式

(1) PPP 封装

在建立 PPTP 控制连接之后，数据就可以在 PPTP 客户端和 PPTP 服务器之间发送了。数据包首先被加密并使用一个 PPP 报头进行封装，即初始 PPP 有效载荷如 IP 数据包、IPX 数据包或 NetBEUI 帧等经过加密后，添加 PPP 报头，封装形成 PPP 帧。PPP 帧使用通用路由封装 GRE 报头进行封装，该报头已针对 PPTP 修改过。然后，GRE 封装的 PPP 帧使用一个 IP 报头进行封装，这个报头包含对应于 PPTP 隧道端点的源和目标 IP 地址。

(2) GRE 封装

通用路由封装 GRE 定义了在一种网络层协议上封装另一种网络层协议的通用方法。GRE 是由 Cisco 和 Net-smiths 等公司于 1994 年提交给 IETF 的，由 RFC 1701 和 RFC 1702 定义。GRE 并不是为 VPN 协议设计的，但 PPTP VPN 中使用 GRE 封装 PPP 帧，这种封装可以称为 GRE 隧道。目前多数厂商的网络设备均支持 GRE 隧道协议。

GRE 的隧道由两端的源 IP 地址和目的 IP 地址来定义，允许用户使用 IP 包封装 IP、IPX 和 AppleTalk 包，并支持全部的路由协议(如 RIP2、OSPF 等)。通过 GRE，用户可以利用公共 IP 网络连接 IPX 网络、AppleTalk 网络，还可以使用保留地址进行网络互连，或者对

公网隐藏企业网的 IP 地址。

PPTP 使用 GRE 的扩展版本来传输用户 PPP 包。这些扩展允许 PAC 和 PNS 之间传输用户数据的隧道提供低层拥塞控制和流量控制。这种机制允许高效使用隧道可用带宽并且避免不必要的重发和缓冲区溢出。经过扩展应用于封装 PPTP 数据包的 GRE 报文格式如图 8.13 所示。

Checksum Present	1b	=0
Routing Present	1b	=0
Key Present	1b	=1
Sequence Number Present	1b	
Strict Source Route Present	1b	=0
Recursion Control	3b	=0
Acknowledgement Number Present	1b	
Flags	4b	=0
Version	3b	=1
Protocol Type	16b	=0x880B
Payload Length	16b	
Call ID	16b	
Sequence Number	32b	
Acknowledgement Number	32b	

图 8.13　用于 PPP 封装的 GRE 消息

- Checksum Present：校验和标志位，当设置为 1 时，表示提供了一个 Checksum 字段。对于 PPTP，该标志总被设置为 0。
- Routing Present：路由标志位，当设置为 1 时，表示提供了一个 Routing 字段。对于 PPTP，该标志总被设置为 0。
- Key Present：Key 字段标志位，当设置为 1 时，表示提供了一个 Key 字段。对于 PPTP，该标志总被设置为 1。Key 字段是 Protocol Type、Payload Length 和 Call ID 字段的组合。
- Sequence Number Present：序列号标志位，当设置为 1 时，表示提供了 Sequence Number 字段。
- Strict Source Route Present：严格源路由标志，当设置为 1 时，表示提供了一个“严格源路由”。对于 PPTP，该标志总被设置为 0。
- Recursion Control：一个用于递归的 3 位标志。对于 PPTP，该字段总被设置为 0。
- Acknowledgement Number Present：一个 1 位标志，当设置为 1 时，表示提供了

Acknowledgement Number 字段。

- Flags：一个用于 GRE 标志的 4 位字段。对于 PPTP，该字段总被设置为 0。
- Version 一个用于表示 GRE 报头版本的 3 位字段。对于 PPTP，该字段总被设置为 1。
- Protocol Type：一个用于存储 GRE 有效负载(payload)的 EtherType 值的 16 位字段。对于 PPTP，该字段总被设置为 0x880B，即 PPP 帧的 EtherType 值。
- Payload Length：一个用于表示 GRE 有效负载长度的 16 位字段。
- Call ID：一个用于表示 PPTP 隧道标识符的 16 位字段。对于 PPTP 连接，Call ID 字段有两个不同的值。一个值用于 PPTP 客户端发送的数据，另一个值用于 PPTP 服务器发送的数据。
- Sequence Number：一个用于表示这个数据包的序列号的 32 位字段。
- Acknowledgement Number：一个 32 位字段，用于表示这个隧道接收的某个 GRE 封装的数据包的最高序列号。

(3) 数据链路层封装

数据链路层封装是 IP 数据包多层封装的最后一层，依据不同的外发物理网络再添加相应的数据链路层报头和报尾。例如，如果 IP 数据包在以太网上传输，则用以太网报头和报尾对 IP 数据包进行数据链路层封装；如果 IP 数据包在点到点网络上传输，如模拟电话网或 ISDN 等，则用 PPP 报头和报尾对 IP 数据包进行数据链路层封装。

(4) PPTP 数据包的接收处理

PPTP 客户机或 PPTP 服务器在接收到 PPTP 数据包后，将做如下处理。

- 处理并去除数据链路层报头和报尾。
- 处理并去除 IP 报头。
- 处理并去除 GRE 和 PPP 报头。
- 如果需要的话，对 PPP 有效载荷即传输数据进行解密或解压缩。
- 对传输数据进行接收或转发处理。

8.1.4 L2F/L2TP

L2F(layer 2 forwarding)由 Cisco 提出，RFC 2341 定义。它可用于传输链路层数据包，如 PPP/HDLC/SLP 等。和 PPTP 不同，L2F 可通过多种载体传输，如 ATM、FR 和 IP 等。L2F 现已逐步被 L2TP 所取代。

如图 8.14 所示，L2F 隧道由 ISP 负责建立，其工作过程描述如下。

① 远程主机通过 PPP 或 SLIP 拨入本地的 ISP 网络。

② L2F 建立一个从 ISP 的 NAS 到私有网络的隧道。该隧道使用面向数据包的协议(如 UDP、帧中继)作为封装协议，以提供端到端的连接。

③ L2F 建立 NAS 到本地网关(home gateway)之间的 PPP 连接。

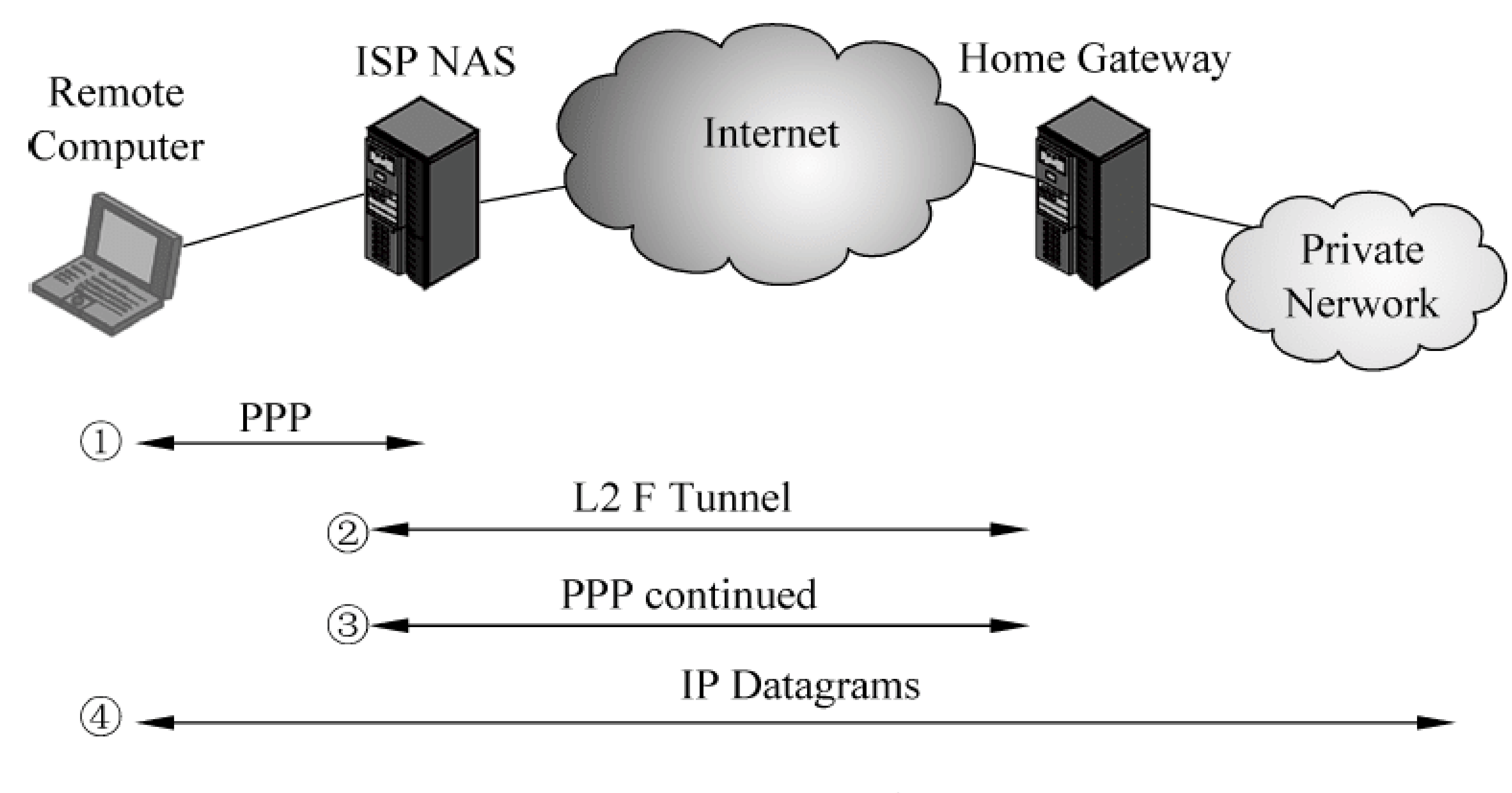


图 8.14 L2F 工作过程

④ PPP 帧通过 IP 协议或其他协议(如 ATM、FR 等)在公共网络上传输。

L2TP(layer 2 tunneling protocol)协议是由 IETF 起草,Microsoft、Ascend、Cisco 和 3COM 等公司参与的二层隧道协议,它结合了上述 L2F 和 PPP 协议的特点,利用公共网络封装 PPP 帧,可以实现和企业原有非 IP 网络的兼容。同时 L2TP 还继承了 PPTP 的流量控制技术,支持 MP(multilink protocol),把多个物理通道捆绑为单一的逻辑信道,并使用 PPP 可靠性发送协议(RFC 1663)实现数据包的可靠传输。L2TP 隧道在两端的 VPN 服务器之间采用密码握手协议 CHAP 来验证对方的身份。L2TP 可以在任何提供面向分组的点对点连接上建立隧道,包括 X.25、帧中继和异步传输模式 ATM。

使用 L2TP 的 VPN 网络逻辑图如图 8.15 所示。当用于 IP 网络环境时,L2TP 同 PPTP 非常相似。一条 L2TP 隧道在一个 L2TP 客户和一个 L2TP 服务器之间建立。客户端可以直接连接到一个 IP 网络或者通过拨号进入一个网络接入服务器来建立 IP 连接。用户数据被封装为 PPP 协议,再使用 L2TP 协议封装后在公共网络上传输。

图 8.16 给出了使用 L2TP 构建 VPN 的网络拓扑。其中,LAC 是 L2TP 访问集中器(L2TP access concentrator),是附属在交换网络上的具有 PPP 端系统和 L2TP 协议处理能力的设备,即 L2TP VPN 客户端。LAC 一般就是一个网络接入服务器 NAS(network access server),它的作用是为用户提供通过 PSTN/ISDN 的网络接入服务。LNS 是 L2TP 网络服务器(L2TP network server),实质就是 PPP 端系统上用于处理 L2TP 协议的服务器软件,即 L2TP VPN 服务器。

在通过 L2TP 构建的 VPN 网络中,LNS 和 LAC 对之间存在着两种类型的连接,一种是隧道(tunnel)连接,它定义了一个 LNS 和 LAC 对;另一种是会话(session)连接,它复用在隧道连接之上,用于表示承载在隧道连接中的每个 PPP 会话过程。

L2TP 数据包封装如图 8.17 所示。用户数据包(典型的,为 IP 数据包)被封装为 PPP 帧,PPP 帧使用 L2TP 协议进行封装,然后再封装为 IP 报文,最后封装为通信子网,如 ATM、FR 等协议的数据包。

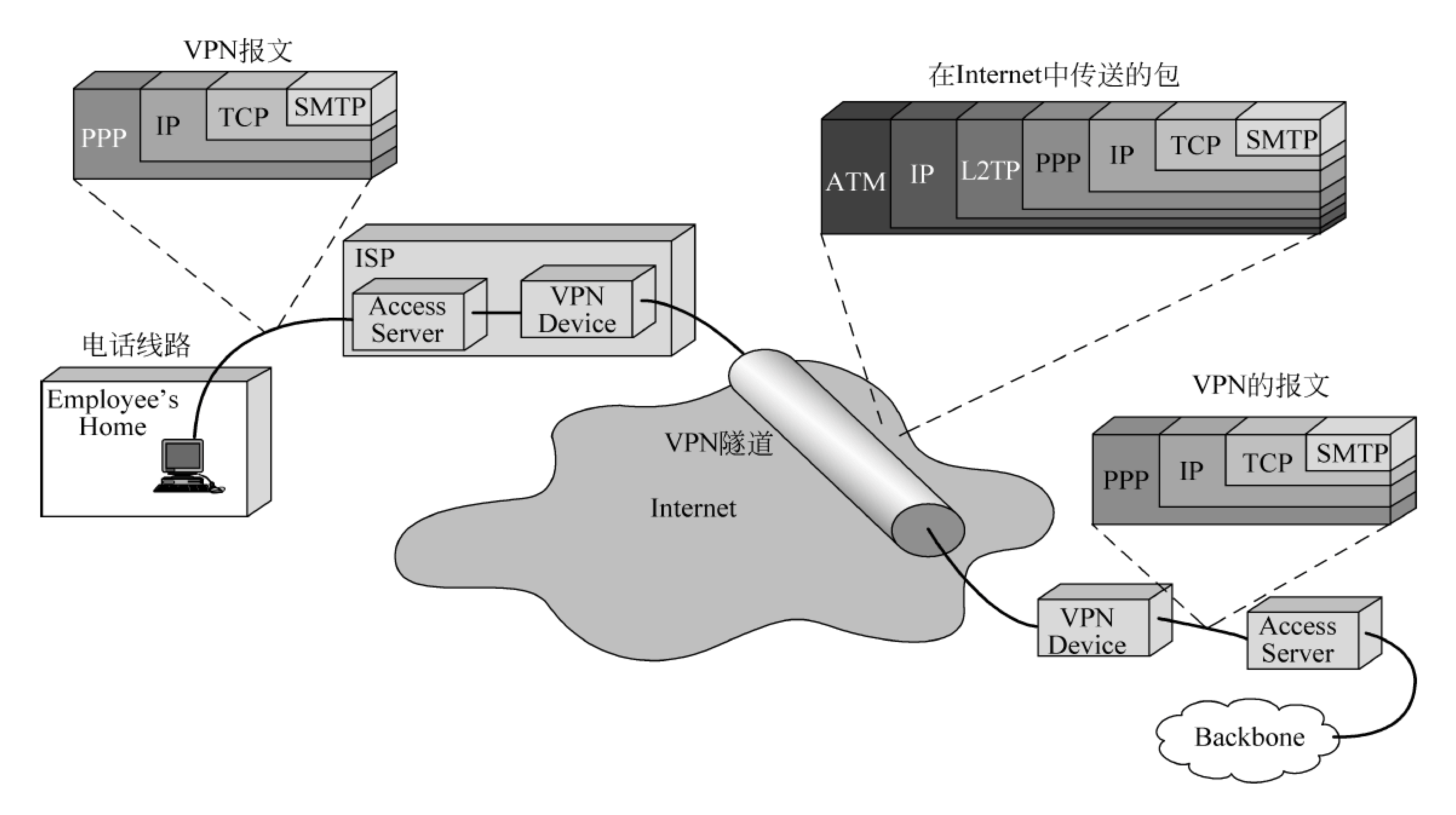


图 8.15 L2TP VPN 的逻辑网络

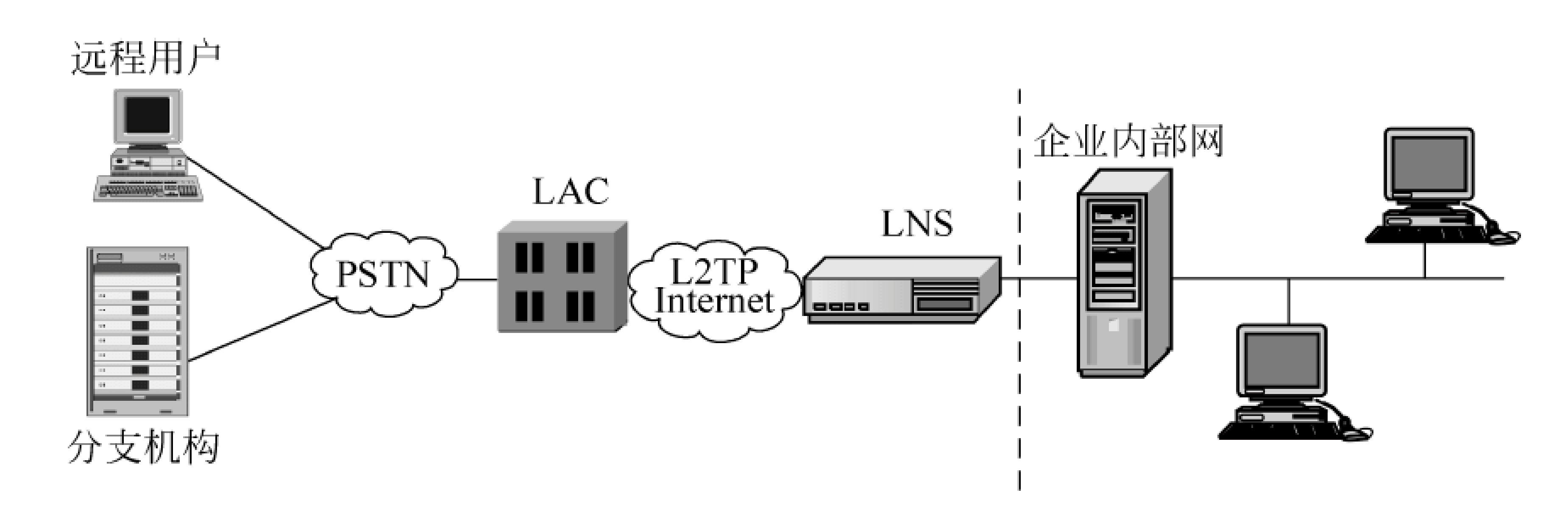


图 8.16 L2TP VPN 网络拓扑

IP/ATM/FR 报头	UDP 报头	L2TP 报头	PPP 报头	IP/IPX 报文

图 8.17 L2TP 封装

L2TP 消息格式如图 8.18 所示。L2TP 连接的维护及 PPP 数据的传送都是通过 L2TP 消息的交换来完成的,这些消息再通过 UDP 的 1701 端口承载于 TCP/IP 之上。

12												16	32 位
T	L	X	X	S	X	O	P	X	X	X	X	VER	Length
Tunnel ID													Session ID
Ns(opt)													Nr(opt)
Offset Size(opt)													Offset Pad(opt)

图 8.18 L2TP 消息格式

L2TP 消息分为两类：一种是控制消息，另一种是数据消息。控制消息用于隧道连接和会话连接的建立与维护，其中的参数用 AVP(attribute value pair)值对来表示，使得协议具有很好的扩展性。同时在控制消息的传输过程中还应用了多种机制来保证 L2TP 传输的可靠性，如消息丢失后重新传送和定时检测通道连通性等。数据消息用于承载用户的 PPP 会话数据包，L2TP 数据消息的传输不采用重新传输机制，所以它无法保证传输的可靠性，但这一点可以通过上层协议如 TCP 等得到保证。此外，数据消息的传输可以根据应用的需要，灵活地采用流量控制或非流量控制机制，甚至可以在传输过程中动态地使用消息序列号，从而动态地激活消息顺序检测和流量控制的功能。

- T：表示消息类型。数据信息为 0；控制信息为 1。
- L：当设置该字段时，说明 Length 字段存在，表示接收数据包的总长。对于控制信息，必须设置该值。
- X：为扩展预留使用。在导出信息中所有预留位被设置为 0，导入信息中该值被忽略。
- S：如果设置 S 位，那么 Nr 字段和 Ns 字段都存在。对于控制信息，S 位必须设置。
- O：当设置该字段时，表示在有效负载信息中存在 Offset Size 字段。对于控制信息，该字段值设置为 0。
- P：如果 Priority(P)位值为 1，表示该数据信息在其本地排队和传输中将会得到优先处理。
- VER：版本信息。
- Length：信息总长，包括头、信息类型 AVP 及与特定控制信息类型相关的 AVP。
- Tunnel ID：识别控制信息应用的 Tunnel。如果对等结构还没有接收到分配的 Tunnel ID，那么 Tunnel ID 必须设置为 0。一旦接收到分配的 Tunnel ID，所有后续数据包必须和 Tunnel ID 一起被发送。
- Sesslon ID：识别控制信息应用的 Tunnel 中的用户会话。
- Nr：期望在下一个控制信息中接收到的序列号。
- Ns：数据或控制信息的序列号。
- Offset Size & Pad：规定通过 L2F 协议头的字节数，协议头是有效负载数据起始位置。Offset Padding 中的实际数据并没有定义。

L2TP 继承了 PPP 的所有安全特性，可以选择多种身份验证机制(如 CHAP、密码验证和 PAP 等)，还可以对隧道端点进行验证，这使得通过 L2TP 所传输的数据更加安全。同时，L2TP 还可以根据特定的网络安全需求，采用隧道加密、端对端数据加密或应用层数据加密等方案来提高数据的安全性。L2TP 协议规范并没有包含加密或者管理用于加密的密钥过程。和 PPTP 相同，L2TP 可以使用 IPSec 来完成 IP 环境下的数据加密和密钥管理。

PPTP 和 L2TP 的比较如表 8.1 所示。

表 8.1 PPTP 和 L2TP 的比较

	PPTP	L2TP
对公共网络的要求	IP	IP、帧中继、X.25、ATM
可建隧道的数量	单一隧道	多条隧道
压缩包头时系统的开销	6 字节	4 字节
隧道验证	不提供	支持
传输协议	TCP	UDP
其他性能		提供差错和流量控制

此外,PPTP 协议使用专门的控制信息即 PPTP 消息对 PPTP 连接以及隧道进行控制和管理,然后使用 GRE 和 PPP 封装用户数据信息。也就是说,在 PPTP VPN 中,控制信息和数据信息采用不同的协议和封装方法,而在 L2TP VPN 中,控制信息和用户数据信息都采用同一种消息即 L2TP 封装,只是消息类型不同。

如前所述,除二层隧道协议 PPTP、L2F 和 L2TP 外,IPSec 也可以用来构建 VPN,即 IPSec VPN,这种 VPN 隧道也称为 IPSec 隧道。IPSec VPN 具有 IPSec 的强大安全特性,包括认证、消息完整性和机密性服务等。而 PPTP 和 L2TP 则需要依赖 IPSec 或其他加密方法才能获得机密性。

同时,IPSec 基于 PKI 技术,它可以为路由器之间、防火墙之间或者路由器和防火墙之间提供经过加密和认证的通信,并可使用证书进行身份鉴别。IPSec 的实现会复杂一些,但其安全性比其他协议完善。由于 IPSec 是基于 IP 的三层隧道协议,因此具有较好的通用性,现在 IPSec 协议也已成为实施 VPN 的重要协议。

8.1.5 MPLS VPN

多协议标记交换 MPLS(multi-protocol label switch)是一种数据包的高速转发技术。与传统逐跳 IP 路由转发机制不同,MPLS 网络中的边缘路由器 LER(label edge router)按照一定规则将数据包分类形成等效前传类 FEC(forwarding equal class),然后标记交换路由器 LSR(label switch router)利用信令协议为 FEC 分配一个定长标记,标记产生后即形成一条从源端到目的端的标记交换路径 LSP(label switch path)。中间 LSR 仅根据该标记来转发数据包,不必进行 IP 最长匹配查找。MPLS 将路由和转发机制相分离,提高了分组的转发速度,可降低数据包的传输延迟,加快网络传输速度。MPLS 提供的高速转发、流量工程和 VPN 等能力使得其具有良好的发展前景,MPLS 被称为下一代骨干网络交换平台。

如图 8.19 所示,MPLS 位于传统五层网络参考模型的第二层与第三层协议之间,其上层与下层可以是当前网络中存在的各种协议。

MPLS 在链路层和网络层之间插入一个 32 位的 MPLS 头部,MPLS 包头的格式如图 8.20 所示。

IPv4	IPv6	IPX	Appletalk	
MPLS				
ATM	Frame Relay	Ethernet	PPP	FDDI...

图 8.19 MPLS 和其他层的关系

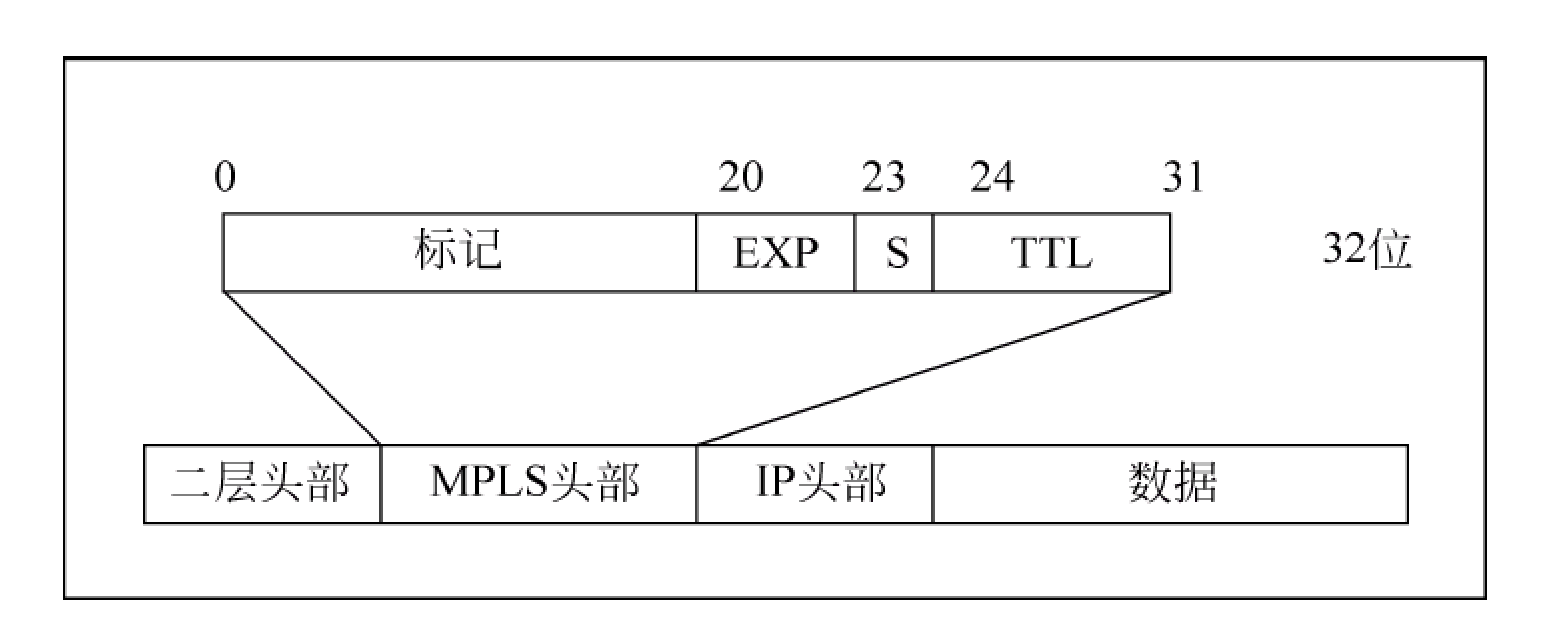


图 8.20 MPLS 数据包格式

其中各字段的说明如下。

20 位的标记字段用来标识标记转发路径。

3 位的 EXP 字段通常用做 CoS(class of service),表明服务级别。

1 位的 S 字段用于标识该 MPLS 标记是否为底层标记。

8 位的 TTL 表明数据生存期,防止环路发生。

图 8.21 给出了一个 MPLS 网络示意图。一个 MPLS 网络的转发设备由标记边缘路由器 LER 和标记交换路由器 LSR 构成。当一个未标记的数据包进入 MPLS 网络时,LER 将根据标记策略为该数据包加上 MPLS 头部。其中 20 位的标记字段作为标记交换路径的索引,隐式地指明该数据包需要经过的路径;中间路由器 LSR 按照该标记值,通过查询标记信息库(label information base,LIB)来确定标记交换路径 LSP,进行 MPLS 数据包的转发。

图 8.19 所示的 MPLS 网络中产生了两条 LSP:其中 LSP1 沿 LER1-LSR1-LSR2-LSR3-LER2,LSP2 沿 LER1-LSR1-LSR2-LSR4-LSR3-LER2。当标记包到达出口路由器 LER2 时,MPLS 包头被去除,数据包被恢复成原始数据包,并根据路由表转发到目的地。

由于路由处理和交换效率的提高,网络延迟已经缩短,而网络的可伸缩性增加了。同时,MPLS 是基于标记的 IP 路由选择方法,这些标记可以被用来代表逐跳式或者显式路由,指明服务质量(quality of service,QoS)、虚拟专网及影响一种特定类型的流量(或一个特殊用户的流量)在网络上的传输方式等各类信息。

对于到达同一目的地的 IP 包,MPLS 可根据其服务质量的要求建立不同的转发路径,以达到其对传输质量的要求。同时,通过对特殊路由的管理,还能有效地解决网络中的负载均衡和拥塞问题。当网络中出现拥塞时,MPLS 可实时建立新的转发路由来分散流量以缓解网络拥塞。

IP 设备和 ATM 设备厂商实现 MPLS 技术是在各自的基础上实施的,对于 IP 设备商,

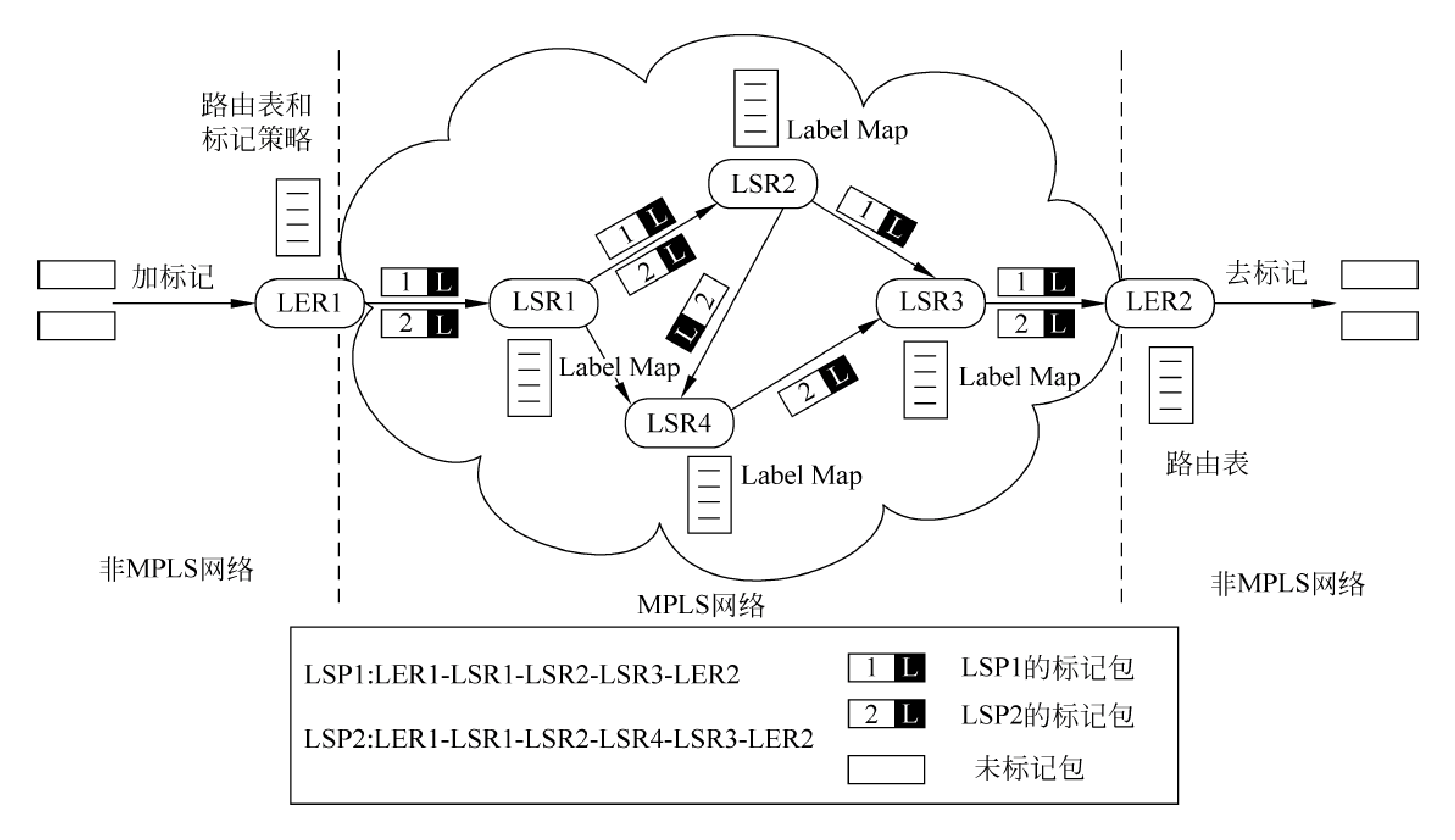


图 8.21　一个 MPLS 网络示意图

它修改了原来 IP 包直接封装在二层链路帧中的规范,在二层和三层 IP 数据包头之间插入一个 20 字节的标签(或标记,label),而 ATM 设备制造商则利用原 ATM 交换机上的 VPI/VCI(virtual path identifier /virtual channel identifier),使用标签代替 VPI/CVI,同时在 ATM 交换机上修引 ATM 信令,引入三层路由和 MPLS 信令,使用路由协议来和其他设备交换三层路由信息。

1. MPLS VPN 体系结构与工作原理

利用 MPLS 技术可以方便地实现 VPN。三层 MPLS VPN 又称 BGP MPLS VPN,是一种基于路由方式的 MPLS VPN 解决方案。IETF RFC 2547 中对该技术做了规定。三层 MPLS VPN 的网络结构,主要由 PE(provider edge device,运营商边缘设备)、P(provide device,运营商设备)和 CE(customer edge device,用户边缘设备)3 种设备组成。下面简要介绍上述设备所实现的功能。

MPLS VPN 体系结构如图 8.22 所示。用户工作场地通过客户边缘设备 CE 路由器接入 MPLS 骨干网络的边缘设备 PE 路由器中。每个工作场地可包含一个或多个 VPN,在 PE 路由器中为每个 VPN 建立一个虚拟路由器,而且每个虚拟路由器都有各自的路由表和转发表,称为 VRF(VPN routing and forwarding)。通过 VRF,一个 VPN 的路由信息不会扩散到其他 VPN 中。利用边界网关协议(border gateway protocol,BGP)和内部网关协议(interior gateway protocol,IGP)在作为 BGP 对等实体的进/出口 PE 路由器之间建立 LSP

隧道,这些隧道可以隔离不同的 VPN,中间 P 路由器仅进行标记数据包的转发,而无法获知网络中的 VPN 信息,从而保证 VPN 数据的不透明性。

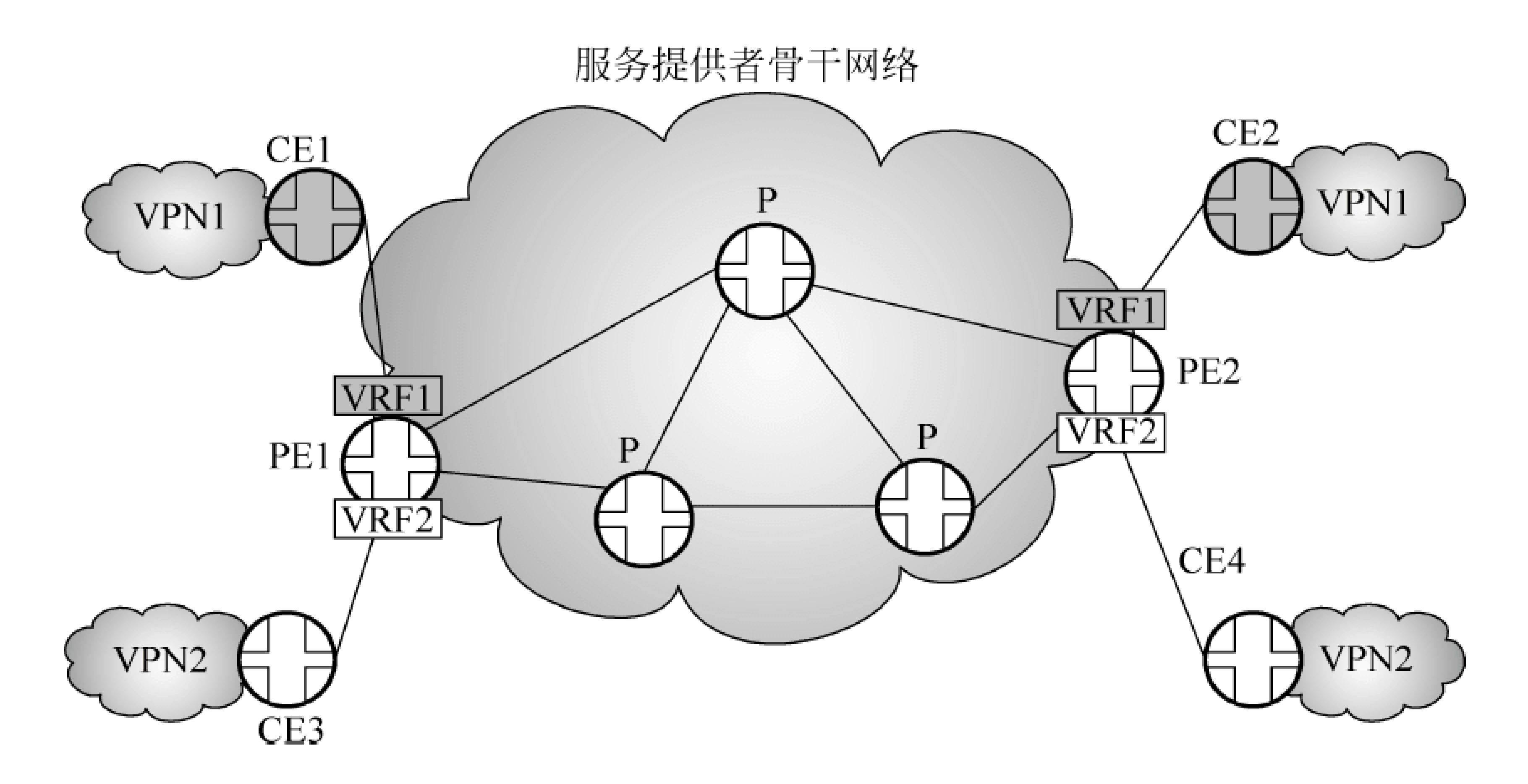


图 8.22 MPLS VPN 网络

在图 8.22 的 PE1 中建立了两个 VRF,分别为 VRF1 和 VRF2,这两个 VRF 分别连接不同的物理设备 CE,即 VRF1 和 CE1 及 VPN1 相连,VRF2 和 CE2 及 VPN2 相连。CE1 将其 VPN1 的路由信息通过内部网关协议 IGP 和 VRF1 交换,然后这种内部路由信息通过 BGP 在骨干网络上传输并发布至 PE2 中,PE2 再将 VPN1 的路由信息通过内部网关协议发布至 CE2。VPN2 的路由信息通过同样的方式发布至远端的 CE4 中,但不会扩散至 CE2 中。因此,通过 BGP、IGP 及虚拟路由器的使用,不同 VPN 之间实现了路由隔离,相同 VPN 之间可以通信。

在 VRF 中定义的和 VPN 业务有关的两个重要参数是 RD(route distinguisher)和 RT(route target)。RD 和 RT 的长度都是 64 位。有了虚拟路由器就能隔离不同 VPN 用户之间的路由,也能解决不同 VPN 之间 IP 地址空间重叠的问题。

正常的 BGP4 协议只能传递 IPv4 的路由,由于不同 VPN 用户具有地址空间重叠的问题,必须修改 BGP 协议。BGP 最大的优点是扩展性好,可以在原来的基础上再定义新的属性,通过对 BGP 修改,把 BGP4 扩展成 MP-BGP。在 MP-IBGP 邻居间传递 VPN 用户路由时打上 RD 标记,这样 VPN 用户传来的 IPv4 路由转变为 VPNv4 路由,这样保证 VPN 用户的路由到了对端的 PE 上,能够使对端 PE 区分开不同但又有地址空间重叠的 VPN 用户路由。

2. MPLS VPN 路由及转发分析

下面使用实例对 MPLS VPN 的路由隔离、网络配置及数据包转发进行分析。

如图 8.23 所示,在 PE1、PE2 和 PE3 上分别配置 VRF 参数,其中 VPN1 用户的 RD=6500:1,RT=100:1,VPN2 用户的 RD=6500:2,RT=100:2。所有 VRF 可以同时导入和导出所定义的 RT。

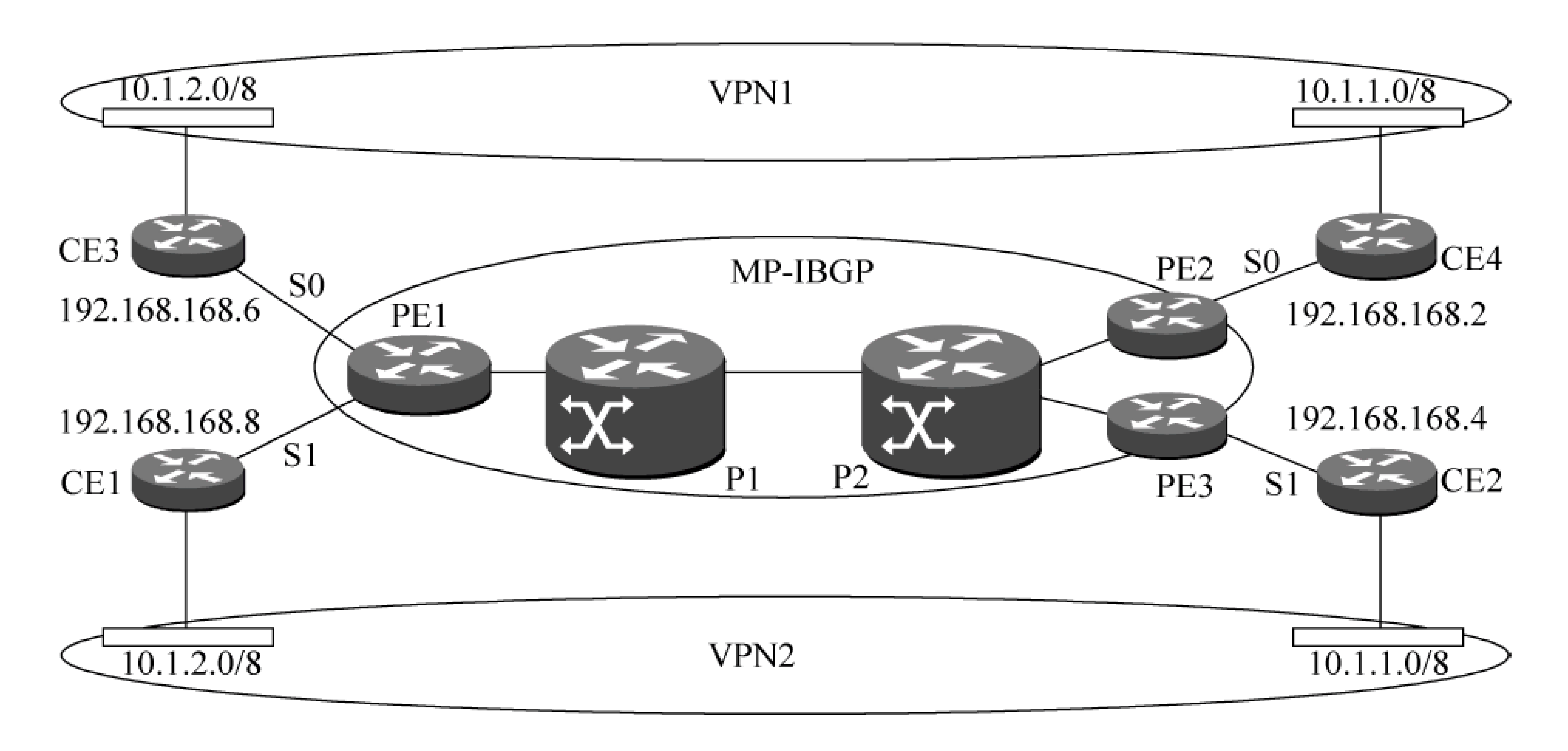


图 8.23 MPLS VPN 网络配置示例

以 PE2 为例，PE2 从接口 S0 上获得由 CE4 传来的有关 10.1.1.0/8 的路由，PE2 把该路由放置到和 S0 有关的 VRF 所管辖的 IP 路由表中，并且分配该路由的本地标签，注意该标签是本地唯一的。通过路由重新发布机制把 VRF 所管辖的 IP 路由表中的路由重新发布到 BGP 表中，此时通过参考 VRF 表的 RD、RT 参数，把正常的 IPv4 路由变成 VPNv4 路由，如 10.1.1.0/8 变成 6500：1 的 10.1.1.0/8，同时把导出(export)RT 值和该路由的本地标签值等属性全部加到该路由条目中去。通过 MP-IBGP 会话，PE2 把这条 VPNv4 路由发送到 PE1 处，PE1 收到了两条有关 10.1.1.0/8 的路由，其中一条是由 PE3 发来的。由于 RD 的不同，导致该两条路由没有可比性。MP-BGP 接收到这两条路由后的后继工作是：去掉 VPN4 路由所带的 RD 值，使之恢复 IPv4 路由原貌，并且根据各 VRF 配置的允许导入(Import)RT 值，把 IPv4 导入到各个 VRF 管辖的路由表和 CEF 表中，也就是说带有 RT＝100：1 的 10.1.1.0/8 的路由导入到 VRF1 所管的路由表和 CEF 表中，带有 RT＝100：2 的 10.1.1.0/8 的路由导入到 VRF2 所管辖的路由表和 CEF 表中。再通过 CE 和 PE 之间的路由协议，PE 把不同的 VRF 管辖的路由表内容通告给各自的相联的 CE 中去。

目前 PE 和 CE 之间可支持的路由协议只有 BGP、OSPF、RIP2 或者静态路由 4 种。

如图 8.24 所示，两层标记转发过程描述如下。

① CE1 接收到发往 10.1.1.1 的 IP 数据包，查询路由表，把该 IP 数据包发送到 PE1。

② PE1 从 S1 口上收到 IP 数据包后，根据 S1 所在的 VRF，查询对应的 CEF 表，数据包打上标签 8，注意该标签就是通过 MP-BGP 协议传来的。PE1 继续查询全局 CEF 表，获知要把数据发往 10.1.1.1，必须先发送到 PE2。而要发送到 PE2，则必须打上由 P1 告知的标签 2。所以该 IP 包被打上了两个标签。

③ P1 接收到标签包后，分析顶层的标签，把顶层标签换成 4，继续发送到 P2。

④ P2 和 P1 一样做同样的操作，由于为顶层标签，P2 去掉标签 4，直接把只带有一个标

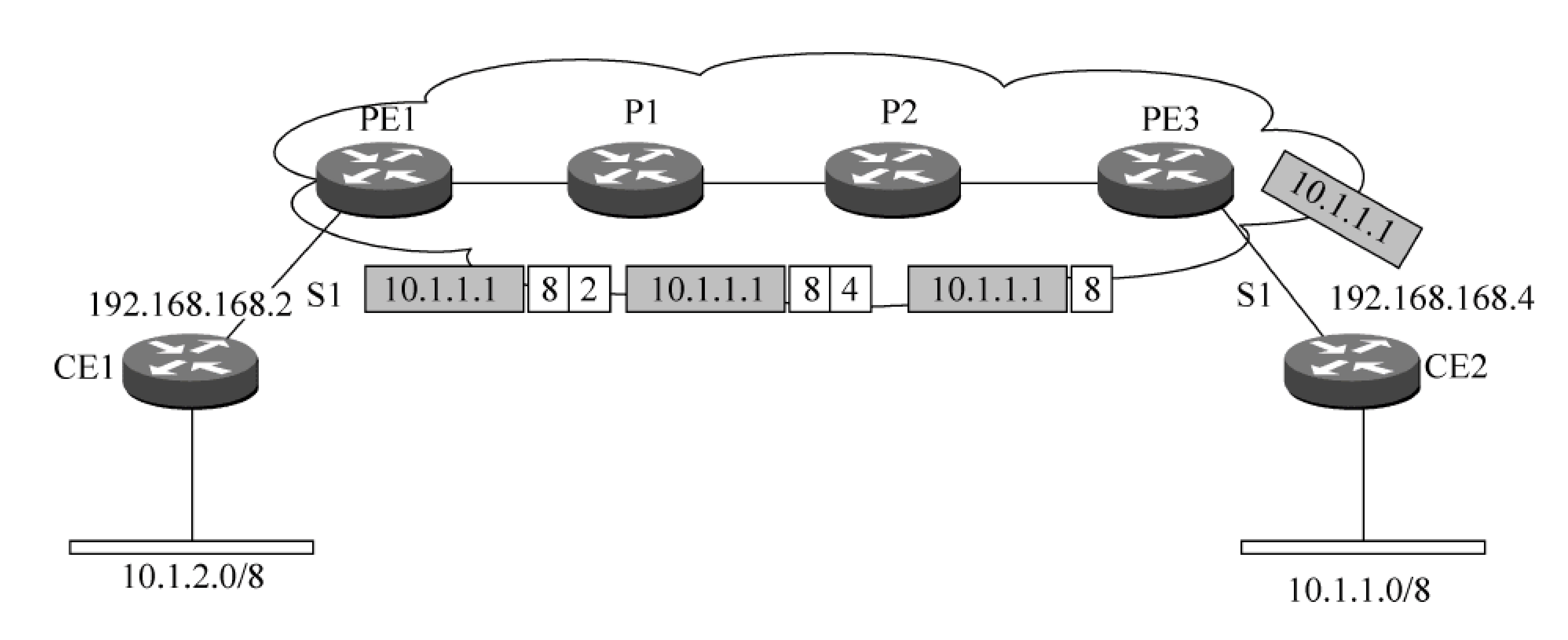


图 8.24 MPLS VPN 两层标记转发机制

签的标签包发送到 PE2。

⑤ PE2 收到标签包后，分析标签头，由于该标签 8 是它本地产生的，而且是本地唯一的，所以 PE2 很容易查出带有标签 8 的标签包应该去掉标签，恢复 IP 包原貌，从 S1 端口发出。

⑥ CE2 在获得 IP 数据包后，进行路由查找，把数据发送到 10.1.1.0/8 网段上。

3. MPLS VPN 实施案例

某公司总部下有 20 个分公司，建有网管、业务综合服务系统 BOSS 及办公自动化 OA 等业务系统。这些业务系统分属不同的部门维护和管理，因此每个系统都各自建有自己的网络。各业务系统所有的业务服务器及核心设备均集中在省公司，各分公司以客户端的形式访问省公司的资源。

改造前，由于不同的业务使用不同的业务支撑网，因此存在多网并存现象。多网并存所带来的问题是：网络资源分散，部分网络资源利用率低，部分网络不能满足日益增长的业务需求；各业务系统之间实现了互联互通，但无法进行有效的安全隔离，安全性较差。

为了改变这种状况，满足企业业务支撑网络整体长期发展的需要，该公司采用 MPLS VPN 技术对现有网络进行了改造。在全公司建立统一的 MPLS 骨干网络来承载公司所有内部业务，不同的业务系统通过划分 VPN 来实现互访与隔离。在分公司和省公司均部署相应的 PE 设备，分公司的 PE 设备用来连接分公司的各业务系统网络，省公司的 PE 设备用来连接各业务系统的服务器。CE 设备则由原来省公司及分公司各业务系统的汇聚路由器来担任。

改造后的网络如图 8.25 所示。在省公司部署两套 P 设备和 PE 设备，在每个分公司分别部署两套 PE 设备，整个骨干网络实现了双平面主、备份。分公司到省公司主用链路速率为 155Mb/s，备用链路速率为 8×2Mb/s。

在 VPN 规划方面，针对不同的业务系统划分了不同的 VPN。全网共划分了 BOSS、网

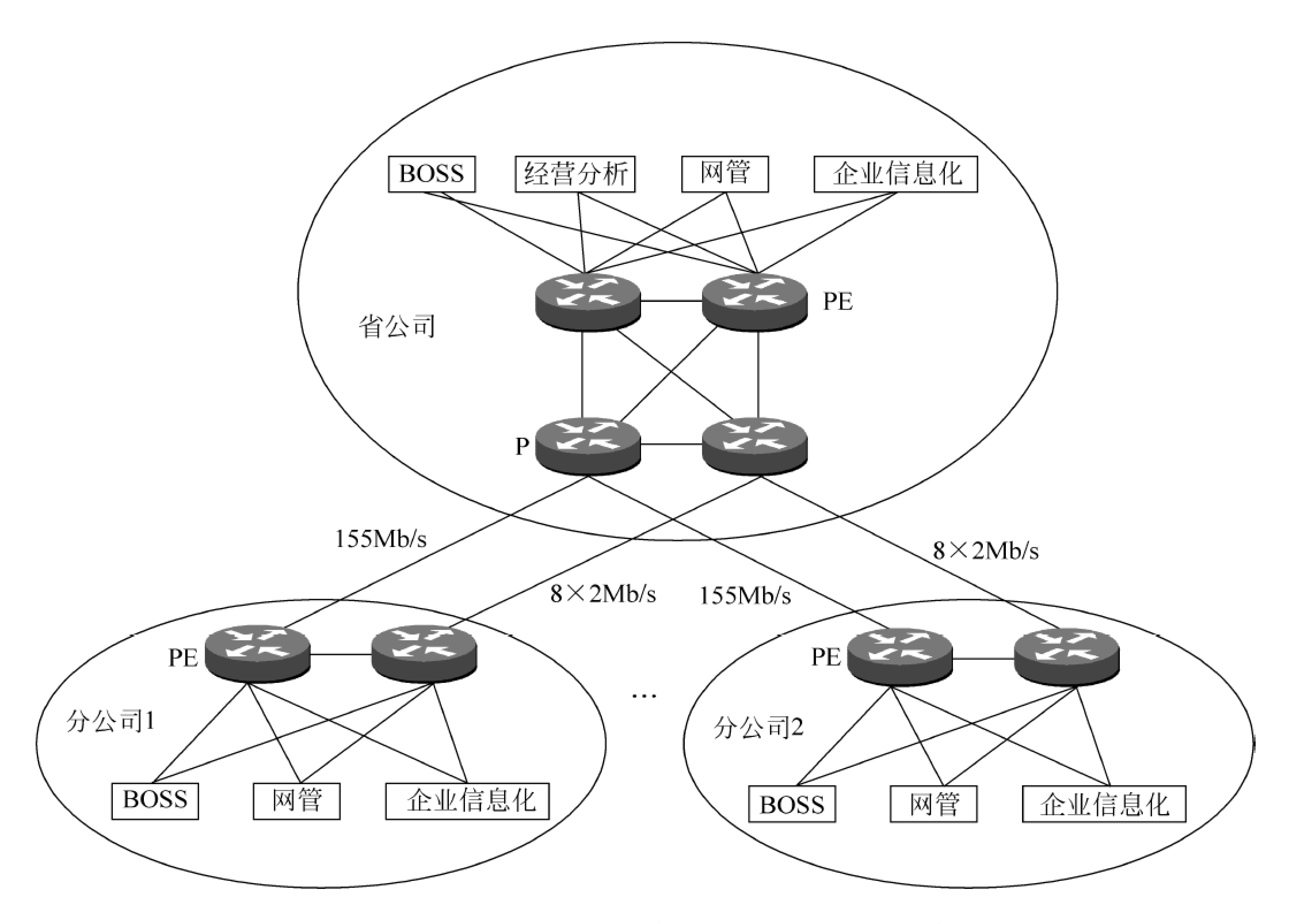


图 8.25　改造后的网络结构(图中未画出 CE 设备)

管、企业信息化和经营分析(只在省中心)4 类 VPN,并通过在 PE 上设置合理的 RT 对 VPN 间的互访与隔离实现了有效控制。所有相同的 VPN 间可以互相访问,所有 VPN 均可访问省中心企业信息化服务器所在的 VPN 及经营分析服务器所在的 VPN。

在整个网络的控制层面,把所有的 P、PE 设备都放在一个域内启用 OSPF 协议,用于标签的分发和建立 LSP。所有的 PE 设备也放在一个域内启用 MP-BGP,用于 VPN 路由的发布和处理。由于 PE 设备间不是采用全连接结构,因此,PE 间需要采用路由反射技术。对于所有的业务而言,分公司都是以客户端的形式访问省公司的资源,因此路由的控制比较简单,实现方法是在 PE 设备和 CE 设备间直接启用静态路由。

由于采用 MPLS VPN 技术组网,因此对于原来各业务系统的 IP 地址规划和各 CE 设备以下网络不需要做任何的改动。在 MPLS 骨干网络建设完成后,只需调整各系统的 CE 设备就可以实现各业务系统的平滑过渡。

与原先的网络相比,采用 MPLS VPN 技术改造后的网络具有以下特点。

(1) 多个业务系统的数据只由一张骨干网络承载,网络结构更加清晰,维护简单。

(2) 安全措施部署简单,业务系统可以进行更加安全的隔离和可控的互访。

(3) 网络扩展性好,当增加新业务系统时不需要建设新的网络,只需增加一个 VPN 即可。不需要针对某个业务系统单独扩容网络带宽,只有当骨干网络平台总带宽不足时才考

虑进行扩容。

(4) 各业务系统统计复用骨干网总带宽,也可以根据各业务系统实际的流量分配带宽,从而合理地使用网络资源,网络资源利用率高。

MPLS VPN 主要用于大型网络的业务隔离,而不适用于远程用户加密访问企业总部信息的情况。它适用于具有以下明显特征的企业:高效运作、商务活动频繁、数据通信量大、对网络依靠程度高、有较多分支机构的大型机构,如 ISP、网络公司、IT 公司、金融业、贸易行业和新闻机构等。企业网的节点数较多,通常将达到几十个以上。例如城域网这样的网络环境,业务类型多样、业务流向流量不确定,特别适合使用 MPLS VPN。

MPLS VPN 的另一个特点是能够在提供 VPN 网络安全性的同时,还具备提供 QoS 的能力,能够实施灵活的控制策略和管理能力。这些优势来自 MPLS 协议本身的特性。

8.1.6 VPN 实施示例

如图 8.26 所示,在最常见的 VPN 配置中,防火墙连接到 Internet,VPN 服务器(可以选择使用 PPTP/L2TP/IPSec 等协议作为 VPN 隧道协议)作为企业内联网(Intranet)资源同周边网络连接。

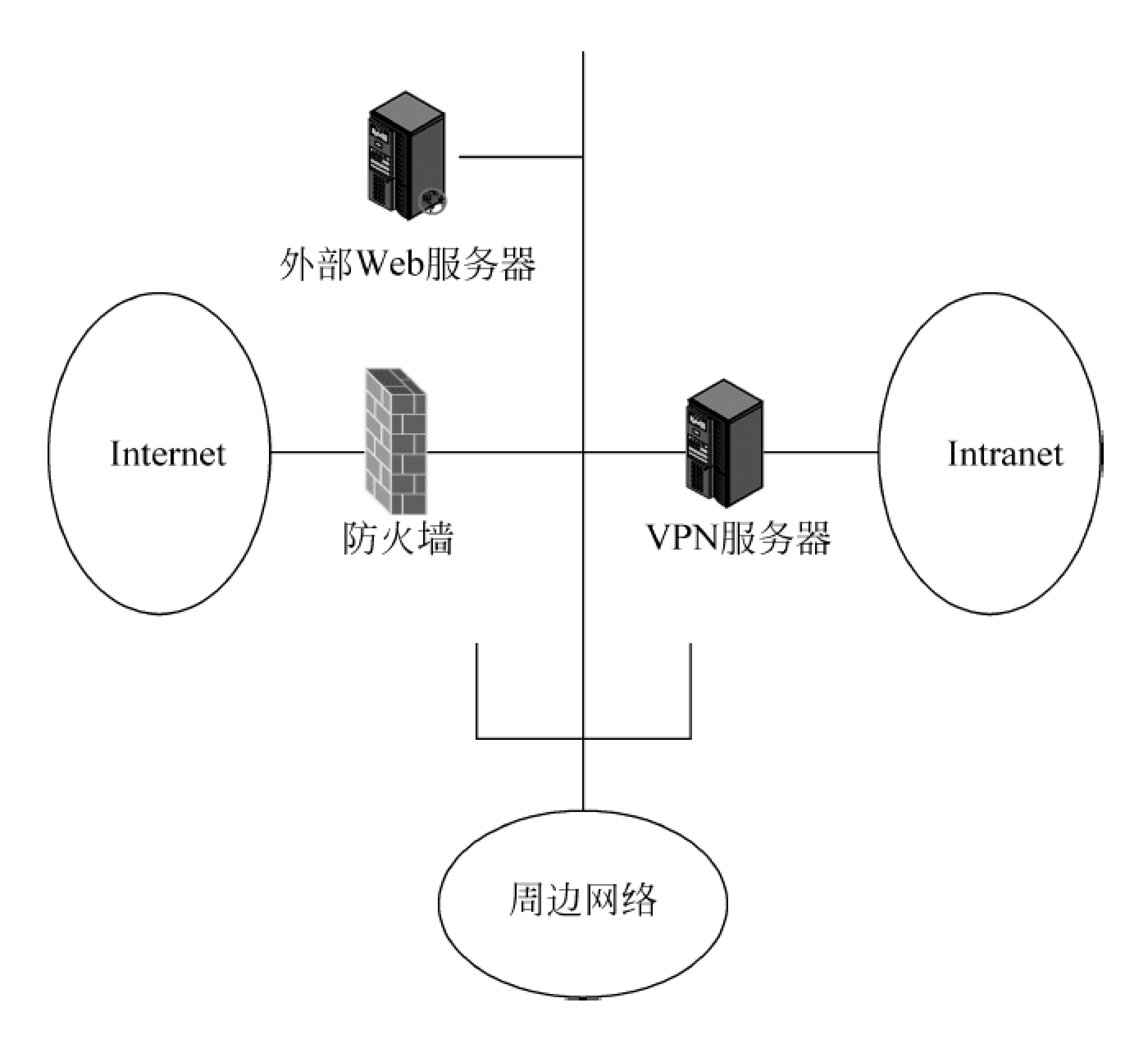


图 8.26 VPN 实施的网络拓扑

VPN 服务器在周边网络和企业内联网上各有一个接口。在此配置中,必须在所属的 Internet 和周边网络接口上通过输入和输出筛选器配置防火墙,从而允许 PPTP 隧道维持流量和通过 PPTP 隧道的数据传输到 VPN 服务器上。

上述 VPN 服务器可以使用微软公司远程访问服务器 RAS(remote access server)实施,这时 RAS 即是一个双穴主机,可负责路由转发及 VPN 服务。同时,RAS 服务器也可以配置为 PPP 拨号服务器,为内部拨号用户提供互联网访问服务。RAS 服务器支持 PPTP/IPSec/L2TP 等多种隧道协议实施 VPN。

上述网络中的 VPN 服务器也可采用 Linux/UNIX 主机实现,只需要在主机上启动路由转发功能,并安装 PPP 和 PPT 软件包且进行适当配置即可。

图 8.27 描述了使用 Modem 进行远程访问的 VPN 连接中,VPN 客户机端数据包在 Windows RAS 网络体系结构中的封装流程。

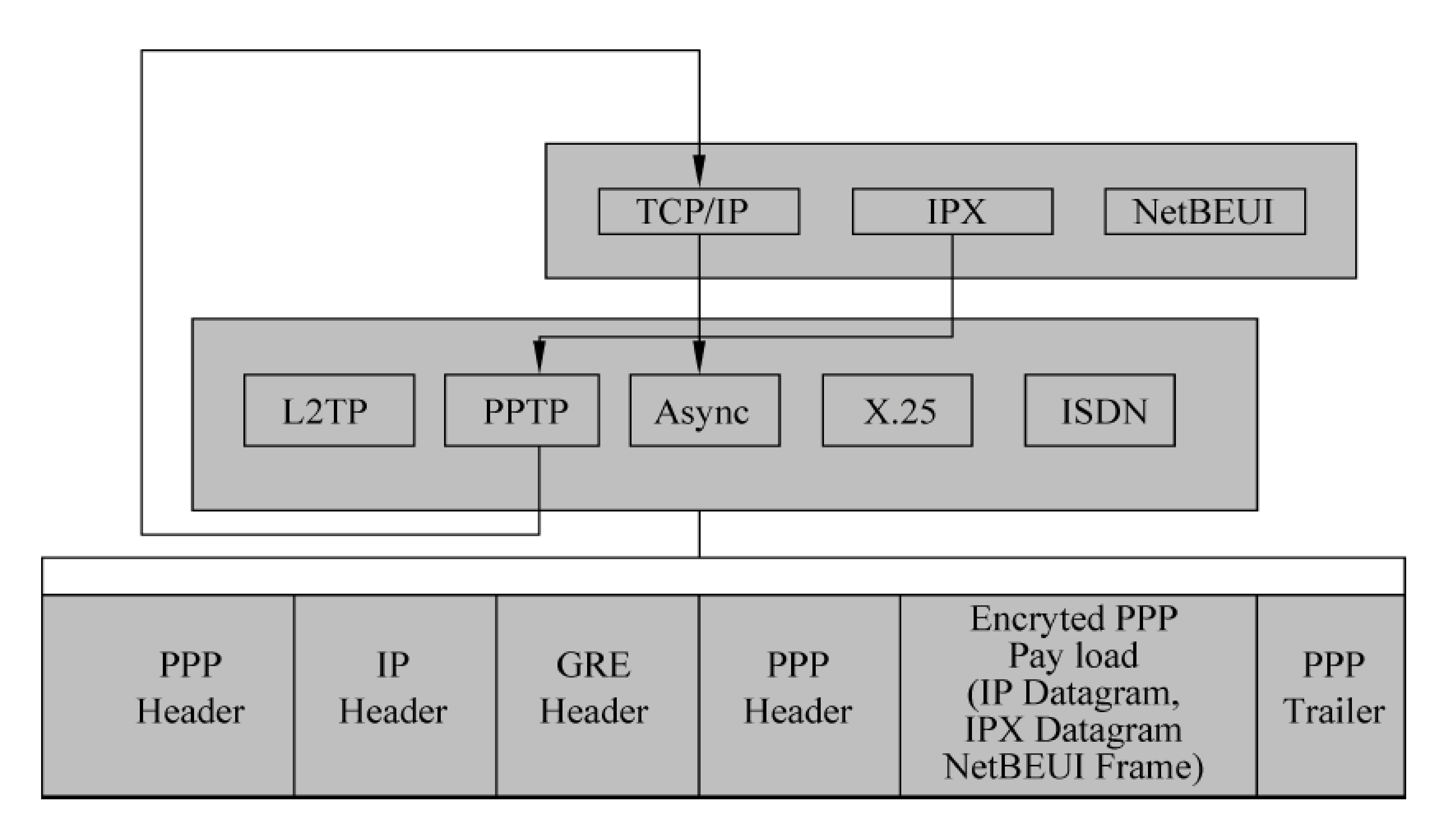


图 8.27 RAS 中 PPTP 数据包的封装流程

① IP 数据包、IPX 数据包或 NetBEUI 帧由各自协议提交给对应于 VPN 连接的虚拟接口。该接口符合网络驱动程序接口规范 NDIS。

② NDIS 将数据包提交给 NDISWAN,由 NDISWAN 负责对数据进行加密、压缩处理后,添加 PPP 报头进行第一步封装。该 PPP 报头仅含一个 PPP 协议标识域,不附加任何帧校正序列 FCS 或其他标记。

③ NDISWAN 将 PPP 帧提交给 PPTP 协议驱动程序,该驱动程序负责在 PPP 帧外添加 GRE 报头进行第二步封装。该 GRE 报头中,Call ID 域的值唯一地标识了一条隧道。

④ TCP/IP 协议驱动程序再对 GRE 报文添加 IP 报头进行第三步封装,封装后再提交给拨往本地 ISP 的拨号连接接口,该接口符合网络驱动程序接口规范 NDIS。

⑤ NDIS 再次将数据包提交给 NDISWAN,NDISWAN 给数据包添加 PPP 报头、报尾进行最后的数据链路层封装。

⑥ NDISWAN 将最终形成的 PPP 帧提交给与拨号硬件相对应的 WAN 微端口驱动程序(例如 Modem 连接中的异步端口)。

8.2 访问控制与安全审计

访问控制(access control)保证只有授权的用户能够在规定的权限内对访问对象进行操作。访问控制的实质是对资源使用的限制：它用于限定主体在网络内对客体所允许执行的动作，即用户在通过鉴别后，还要通过访问控制，才能执行特定的操作。

访问控制技术起源于 20 世纪 70 年代，是为了满足当时管理大型主机系统上共享数据授权访问的需要而使用的。随着计算机技术和应用的发展，特别是网络技术的发展和应用，这一技术的思想和方法迅速应用于信息系统的各个领域。在 30 多年的发展过程中，先后出现了多种重要的访问控制技术，它们的基本目标都是防止非法用户进入系统和合法用户对系统资源的非法使用。为了达到这个目标，访问控制通常以用户身份认证为前提，在此基础上实施各种访问控制策略来控制和规范用户在系统中的行为。

用户、设备或进程对系统的操作，包括访问控制等信息需要进行记录和审计。安全审计系统是事前控制主体对客体的访问行为，并能事后获得直接电子证据，防止行为抵赖的系统。它是信息安全保障系统的重要组成部分。在信息安全领域，从来就没有一种技术可以绝对地保证系统的安全。即使技术在理论上可以很安全，也不可能保证执行人员可以完整无误地执行。因此，无论技术有多先进，审计功能依然非常重要。

8.2.1 访问控制策略

访问控制策略是指实施访问控制所采用的基本思路和方法。目前，常见的访问控制策略包括自主访问控制(discretionary access policies)、强制访问控制(mandatory access policies)和基于角色的访问控制(role-based access policies)三种。

在描述访问控制策略之前，给出访问控制系统的基本概念。

- 主体：访问动作的发起者，即对客体实施动作的实体，例如用户、用户进程等。
- 客体：即被访问对象，计算机系统中所有可控制的资源均可抽象为客体，如文件、设备和内存区数据等。
- 授权：主体对客体所实施的动作需要得到授权，这些授权对于主体可表示为访问权限，对于客体为访问模式。

1. 自主访问控制

自主访问控制随分时系统的出现而产生。其基本思想是：系统中的主体可以自主地将其拥有的对客体的访问权限全部或部分授予其他主体。其实现方法一般是建立系统访问控制矩阵。其中，矩阵的行对应系统的主体，列对应系统的客体，元素表示主体对客体的访问

权限。为了提高系统的性能，在实际应用中常常是建立基于行（主体）或列（客体）的访问控制方法。

基于行的方法是在每个主体上都附加一个该主体可以访问的客体的明细表。根据表中信息的不同可分为三种形式：权能表（capabilities list）、前缀表（porfiles）和密码（password）。

权能表（如图 8.28 所示）决定用户是否可以对客体进行访问及进行何种形式的访问（如读、写、删改和执行等）。一个拥有某种权力的主体可以按一定方式访问客体，并且在进程运行期间访问权限可以添加或删除。

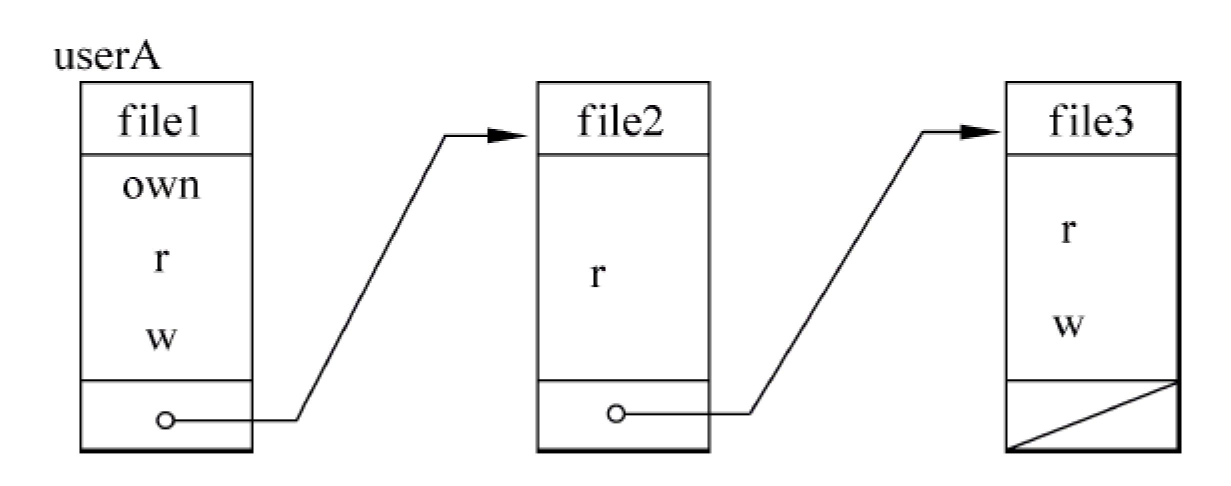


图 8.28　权能关系表

前缀表包括受保护的客体名及主体对它的访问权限。当主体欲访问某客体时，自主访问控制系统将检查主体的前缀是否具有它所请求的访问权。至于密码机制，每个客体（或客体的每种访问模式）都需要一个密码，主体访问客体时首先提供该客体的密码。

DAC 根据用户的身份及允许访问权限决定其访问操作。即只要用户身份被确认后，才可根据访问控制表上赋予该用户的权限进行限制性用户访问，这种访问的灵活性高，被大量地采用（如 UNIX 系统中）。然而也正是由于这种灵活性使信息系统的安全性降低。

DAC 的缺点是访问权的授予是可以传递的。一旦访问权被传递出去将难以控制，访问权的管理是相当困难的，可能带来严重的安全问题。另一方面，DAC 不保护受保护的客体产生的副本，即一个用户不能访问某一客体，但能够访问该客体的备份，这更增加了管理的难度。并且，在大型系统中，主、客体的数量巨大，无论是用哪一种形式的 DAC，所带来的系统的开销都是很大的，效率较低，难以满足大型应用系统的需求。

2. 强制访问控制

MAC 通过无法回避的访问限制来阻止直接或间接的非法入侵。系统中的主、客体都被分配一个固定的安全属性，利用安全属性决定一个主体是否可以访问某个客体。安全属性是强制性地由安全管理员分配的，用户或用户进程不能改变自身或其他主、客体的安全属性。

MAC 的本质是基于格的非循环单向信息流策略，系统中每个主体都被授予一个安全证书，而每个客体被指定为一定的敏感级。访问控制的两个关键规则是：不向上读和不向下写，即信息流只能从低安全级向高安全级流动。任何违反非循环信息流的行为都是被禁

止的。

MAC 中每个用户及文件都被赋予一个访问级别,它是由一个授权机构为主体和客体分别定义固定的访问属性,且这些访问权限不能通过用户来修改。通过定义用户的不同权限,使用户可以访问对应不同安全级别的数据,从而避免了自主访问控制方法中出现的访问传递问题。这种方法具有层次性的特点,高安全级别的权限可访问低级别的数据。

MAC 起初主要用于军方应用,并且常与 DAC 结合使用,主体只有通过 DAC 和 MAC 的检查后,才能访问客体。由于 MAC 对客体施加了更严格的访问控制,因而可以防止特洛伊木马之类的程序偷窃受保护的信息,同时 MAC 对于用户意外泄露机密信息的可能性也有一定的预防能力。但是如果用户恶意泄露信息,则可能无能为力。而且,由于 MAC 增加了不能回避的访问限制,因而可能影响系统的灵活性。另一方面,虽然 MAC 增强了信息的机密性,但不能实施完整性控制,而一些完整性控制策略却可以实现机密性的功能。再者,网上信息更需要完整性,这影响了 MAC 的网上应用。最后,在 MAC 系统实现单向信息流的前提是系统中不存在逆向潜信道。逆向潜信道的存在会导致信息违反规则的流动,而现代计算机系统中这种潜信道是难以去除的,如大量的共享存储器及为提升硬件性能而采用的各种 Cache 等,这给系统增加了安全性漏洞。

3. 基于角色的访问控制

基于角色的访问控制以角色为中介对用户进行授权和访问控制:主体对客体的访问控制权限通过角色实施,即访问权限是针对角色而不是直接针对用户的。RBAC 中,系统安全管理员根据需要定义各种角色,并为其设置合适的访问权限,然后根据用户所担任的工作职责或级别给其分配相应的角色,从而使用户获得相关的权限集。RBAC 利用角色作为桥梁将用户与权限联系起来,而不像传统访问控制技术中那样将访问权限直接分配给用户。根据应用可以给用户分配一些角色,再给角色分配相应的权限;还可以根据应用需要将分配给用户的某些角色撤销,也可以将分配给角色的某些权限撤销。

角色是根据用户在系统中表现的活动性质而确定的,即这种活动性质表明用户担当了一定的角色。角色可以看成是一个表达访问控制策略的语义结构,它可以表示用户承担特定工作的资格,也可以体现某种权利与责任。例如一个银行系统中的角色可以有出纳员、会计师、贷款员和部门经理等,他们的职能不同,拥有对系统的访问权限也不相同。RBAC 根据用户在组织内所处的角色做出访问授权与控制,但用户不能自主地将访问权限传递给他人。例如,在医院里,医生这个角色可以开处方,但他无权将开处方的权力传递给护士。

如图 8.29 所示,引入角色的管理模式后,对资源授权管理过程将分成两个部分,即首先实现访问权限与角色相关联,然后再实现角色与用户相关联,从而实现了用户与访问权限的逻辑分离。用户访问系统时,系统首先检查用户的角色,然后根据角



图 8.29 RBAC 访问控制策略

色拥有的权限决定其对客体的访问能力。

在 RBAC 中,由系统的安全策略决定如何进行权限—角色指派及角色—用户指派等。一个用户可以担当多个角色,一个角色也可以分配给多个用户。

RBAC 作为 DAC 和 MAC 的替代策略在近年来逐渐引起广泛的关注。基于角色访问控制模型的基本思想是让权限与角色相映射。

RBAC 的优势在于它对管理能力的支持。一个 RBAC 系统建立后,主要的管理工作即为分配和取消用户的角色。用户的职责变化时,赋予不同的角色,也就改变了用户的权限。当组织的功能变化或演进时,只需删除角色的旧功能、增加新功能,或定义新角色,而不必更新每一个用户的权限设置。从这个意义上讲,基于角色的访问控制克服了传统 DAC 和 MAC 的不足,可以减少授权管理的复杂性、降低管理开销,并为管理员提供一个较好的实现安全策略的环境。因此,RBAC 在目前大型商业和政府部门的安全应用中开始显示出较大的优势,基于角色的访问控制已经成为传统访问控制策略的发展和补充。

RBAC 的适用范围非常广泛。例如,Windows NT 采用了类似于 RBAC 的访问控制机制。从政府机构到商业应用,RBAC 可以满足许多应用系统的安全需求。另外,RBAC 也非常适用于数据库应用层的安全模型,因为在应用层,角色的逻辑意义更为明显和直接。现有的商品化 RDBMS 系统中,ORACLE8.0 以上和 SYBASE7.0 以上都部分实现了基于角色的访问控制。同时,RBAC 适合对集中的资源进行访问控制,常见于信息集中型应用系统。它适合于“集中控制、集中管理”类型的应用,如电子政务系统等。

1995 年,Ravi Sandhu 等人提出 RBAC 的基本模型,称为 RBAC0 模型。该模型定义了任何支持 RBAC 的技术都必须包含的最小功能子集。1996 年在 RBAC0 模型的基础上加入了角色等级管理,提出了 RBAC1 模型。在 RBAC0 模型中加入了对 RBAC 各种元素的约束,提出了 RBAC2 模型。RBAC1 和 RBAC2 模型都称为高级 RBAC 模型,但它们是不兼容的。为了将这两类模型合并在一起,对 RBAC0 模型进行了改进,形成 RBAC3 模型。RBAC0、RBAC1、RBAC2 和 RBAC3 统称为 RBAC96 模型家族,如图 8.30 所示。

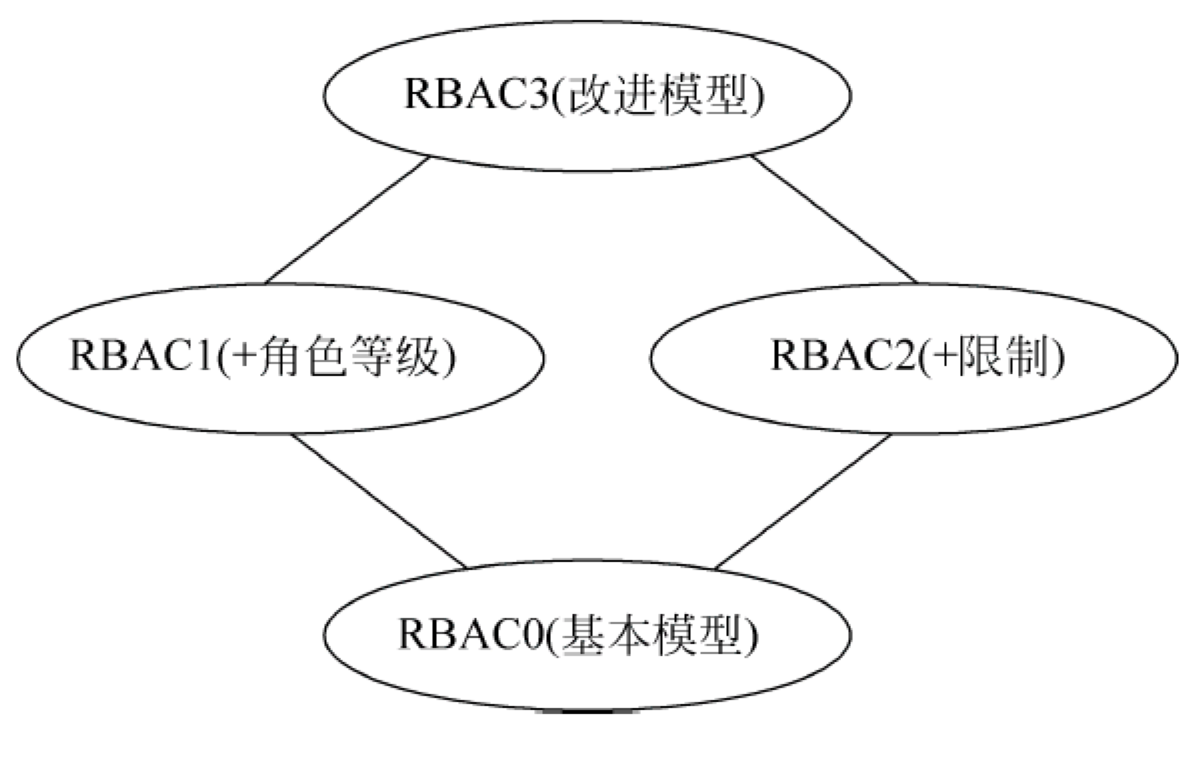


图 8.30　RBAC96 模型

RBAC96 模型家族中的所有模型都有一个共同的特点，它们都假定由一个安全管理员来控制所有的访问控制模型组件。在大型应用系统中，角色、用户和权限数量巨大，对于这些组件的管理将会非常复杂，仅仅依靠人力来控制不仅工作量大，而且很容易造成安全隐患。1997 年，Sandhu 及其他访问控制专家们扩展 RBAC96 模型家族，提出了利用管理员角色来管理角色的思想，发展成为 RBAC97 模型（administrative role based access control）。

RBAC 实质上是一种策略中立的访问控制策略，即它本身并不提供一种特定的安全策略。RBAC 通过配置各种参数（例如文档的安全标志和用户的角色）来实现某种安全策略。从根本上讲，RBAC 仍然属于基于主体、客体的访问控制体系的范畴。在基于主客体访问控制的模型中，基本实体是主体、客体和权限，主体通过权限获得对不同客体的访问能力。

无论哪种访问控制策略，访问控制和授权信息均可以使用如下方法实现。

（1）基于列的访问控制

基于列的自主访问控制对每个客体附加一个它可以访问主体的明细表。其实现有两种形式：保护位（protection bits）和访问控制列表（access control list，ACL）。保护位是对所有的主体指明一个访问模式集合，由于它不能完备地表达访问控制矩阵，因而很少使用。

如图 8.31 所示，访问控制列表可以决定任一主体是否能够访问该客体，它是用在该客体上附加一主体明细表的方法来表示访问控制矩阵的。表中的每一项包括主体的身份和对客体的访问权，访问控制列表是实现自主访问控制的常用方法。

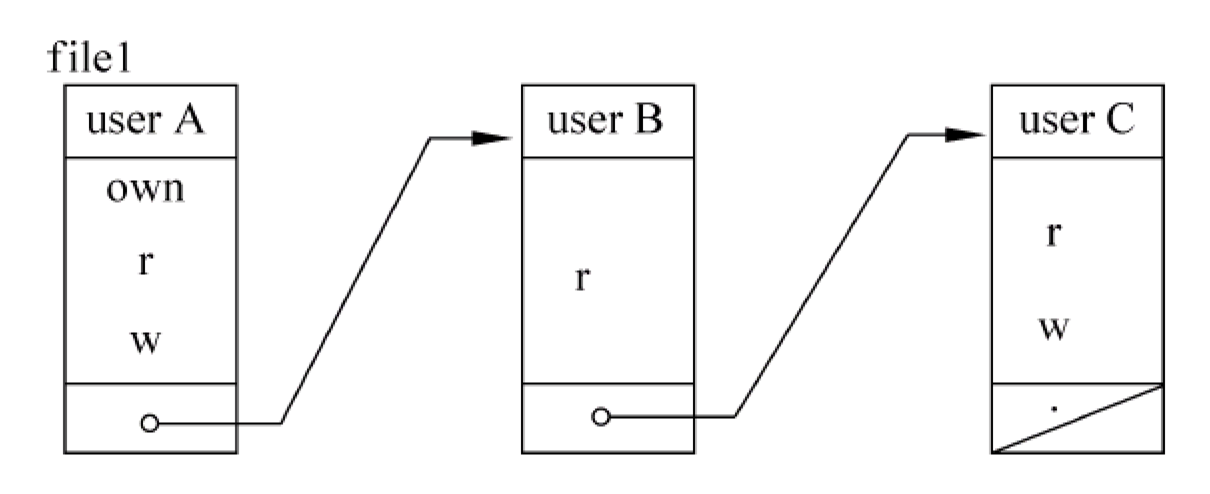


图 8.31 访问控制表

（2）权能表（capabilities lists）

如图 8.28 所示，权能表以主体为索引，为每个主体建立一个 CL（capabilities list），指出对各个客体的访问权限。

（3）权限关系表（authorization relation）

如表 8.2 所示，权限关系表是 ACL 和 CL 的结合，使用关系来表示访问控制矩阵。每个关系表示一个主体对一个客体的访问权限，可以使用关系式数据库来存放这个访问矩阵。这种方式对于主体和客体的授权处理都比较方便，但实现的开销较大。

表 8.2 权限关系表

user A	own	file1
user A	r	file2
user A	w	file3
user B	r	file4
user B	own	file5

8.2.2 访问控制实施模型

访问控制策略决定系统以什么样的方法进行用户的权限划分及授权管理，而没有解决如何实施访问控制的问题。本节讲述访问控制实施的方法。一个访问控制实施模型包括访问控制要素、访问控制策略等。控制要素包括主体、客体和授权。典型的访问控制模型如图 8.32 所示。访问控制可以通过系统中的引用监控器(reference monitor)来实施。

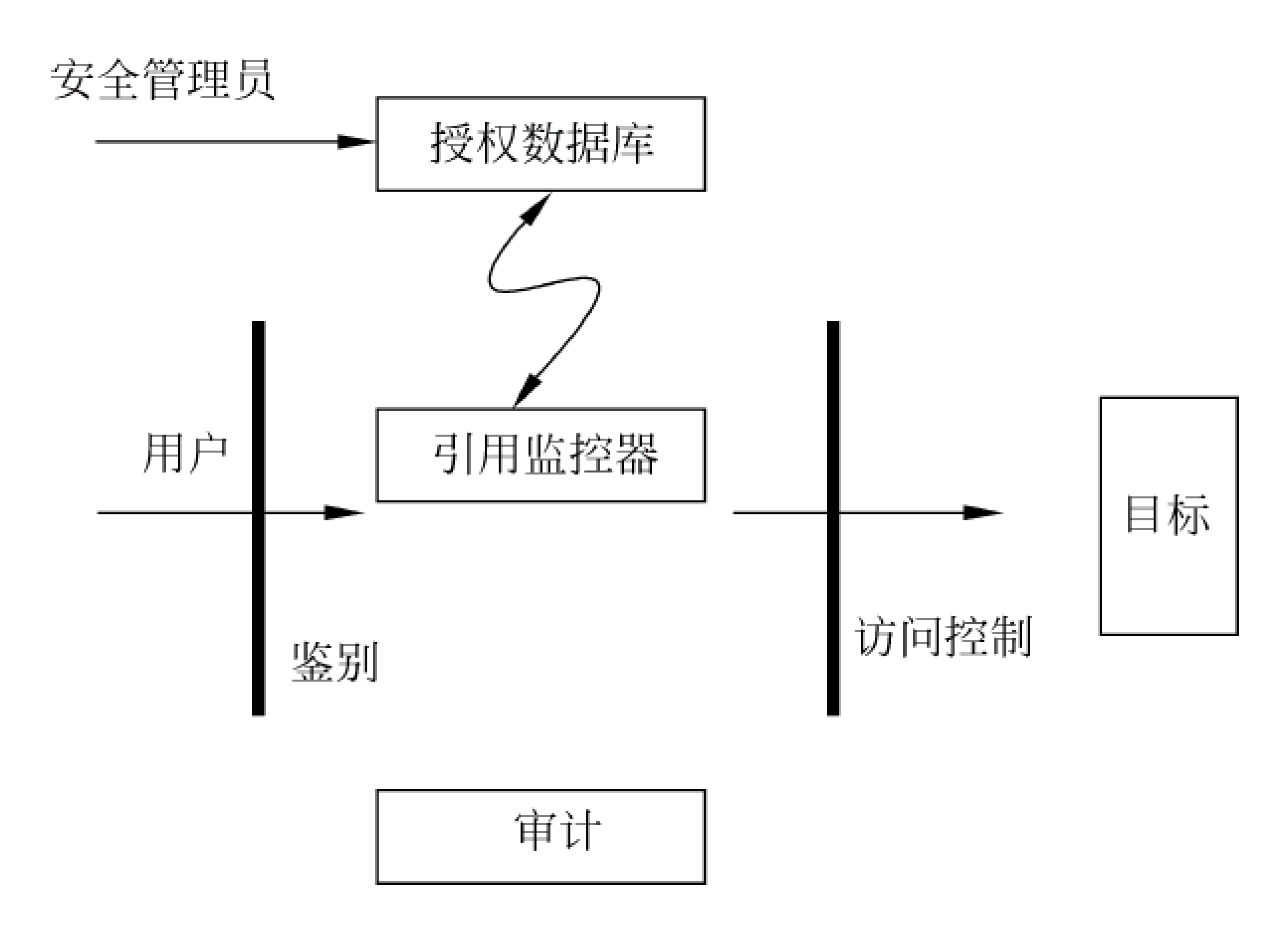


图 8.32 访问控制实施模型

系统中，用户或进程(主体)通过系统鉴别后，在访问目标资源(客体)之前，其访问请求首先被访问控制的引用监控器拦截，监控器检查访问控制策略(可以是 DAC、MAC 或 RBAC 中的一种)，并根据不同的策略实现方式查找授权数据库中的访问控制规则，决定该访问是否被允许。

访问控制规则和授权数据库由安全管理员创建和维护，并且，用户对资源的访问过程及其结果可以被安全审计系统记录。

在企业应用系统或操作系统设计时普遍采用上述模型，即采用监控器拦截用户进程对系统的访问，然后根据访问控制策略和授权信息决定是否允许相应的操作。可以看出，引用

监控器是整个访问控制模型的核心。

在操作系统中,用户、可能含有敏感数据的对象及控制资源访问的部件等共同构成系统环境。操作系统并不是对用户所有的访问资源(例如文件和内存区域的请求)都进行检查。攻击者可能寻找出一些不需要进行检查的访问形式,并能够用它来从用户的文件中窃取客户信息。黑客也可能会在系统中留下一个程序,该程序会随机地覆盖内存区域,从而对其他用户造成严重的破坏。为了解决这些安全问题,可以使用上述访问控制实施模型:当对象被创建时即定义好其他进程对该对象的访问权限,并且,当一个进程试图访问该对象时就执行这些授权规则或策略。

下面给出基于上述模型的访问控制系统的设计示例。图 8.33 为访问控制系统的类图。系统使用引用监控器拦截进程的访问请求。REFERENCE MONITOR 根据访问规则检查进程是否具有对所请求类型的访问权限。访问请求是进程和引用监控器之间的关联类。访问类型为进程和被访问对象之间的关联类。REFERENCE MONITOR 对应多条访问控制规则。

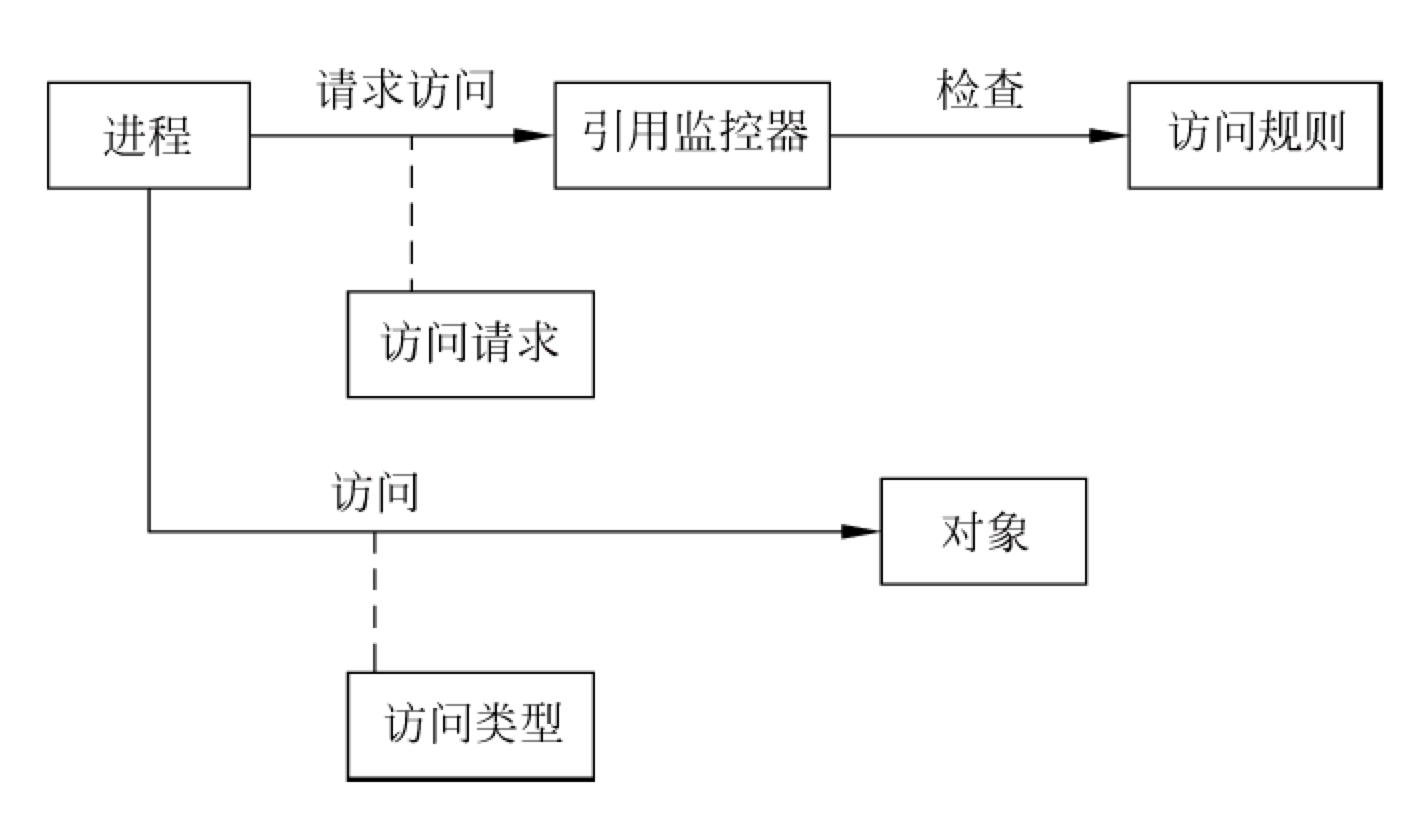


图 8.33　访问控制实施模型类图

图 8.34 描述了上述以引用监控器为中心的访问控制实施模型中安全主体访问安全客体的动态过程。请求被发送给 REFERENCE MONITOR 来检查访问规则。如果允许访问,则执行该访问并且返回结果。返回给主体的是一个句柄或令牌(token),这样主体以后对该安全对象的访问就可以直接进行而不需要进行额外的检查了。

REFERENCE MONITOR 仲裁所有的访问请求。现在所有的请求都要经过检查,因此黑客不能访问未被授权的文件或内存区域。

Windows NT 的安全子系统就是采用上述访问控制模型提供安全保障的。它包含以下 3 个组件。

- 本地安全授权(local security authority,LSA)
- 安全审计管理(security audit management,SAM)
- 安全引用监控器(security reference monitor,SRM)

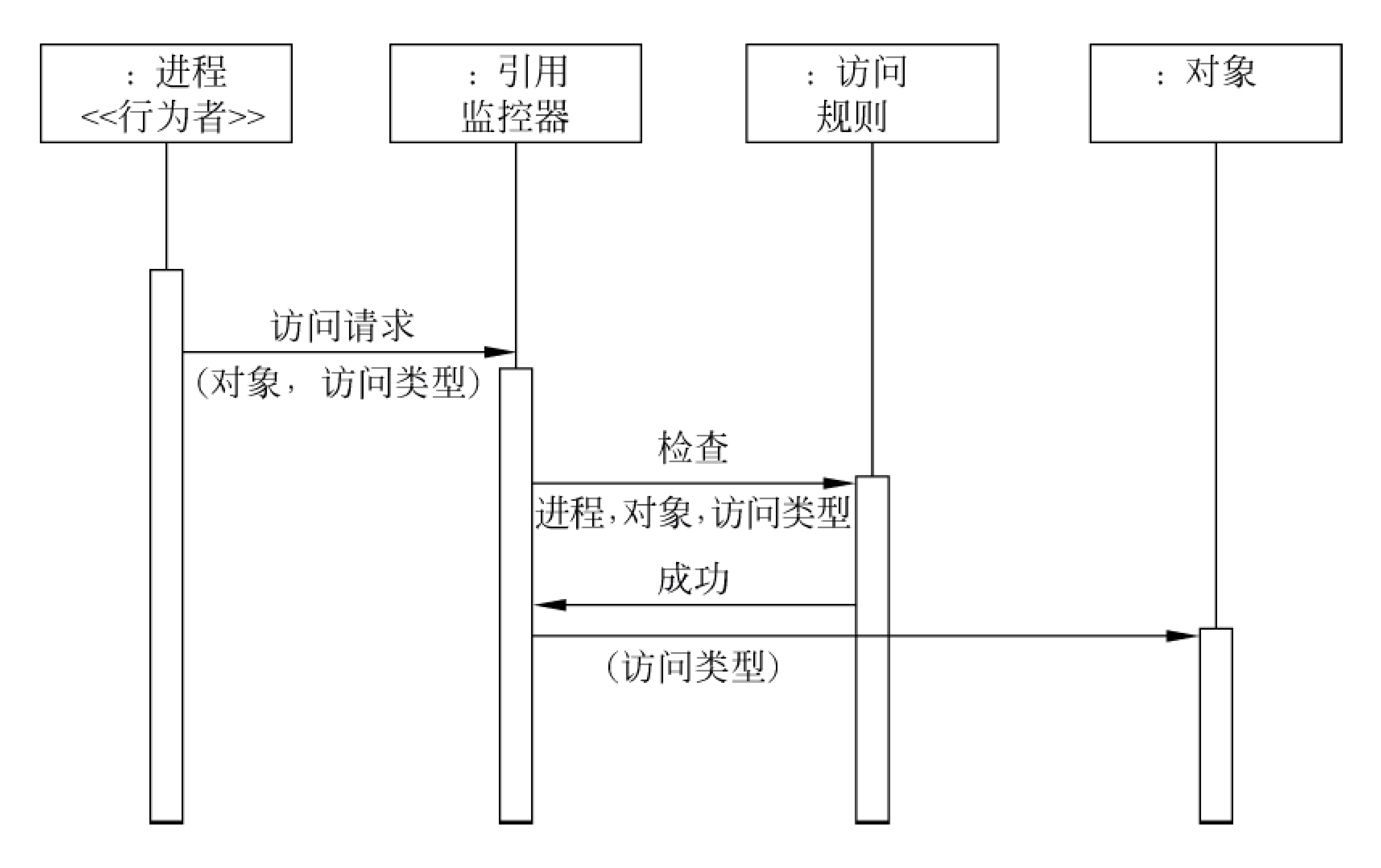


图 8.34　验证访问请求过程的时序图

本地安全授权和安全审计管理两个组件协同来验证用户，并且为用户创建访问令牌。安全引用监控器运行在内核模式下，负责实施访问验证。当主体请求访问一个对象时，系统会把相应文件的安全描述符和保存在用户访问令牌中的安全 ID 进行对比。安全描述符由访问控制条目(access control entry，ACE)组成，这些访问控制条目包含在访问控制列表中。如果对象拥有一个 ACL，安全审计管理会检查 ACL 中的每个 ACE 来决定是否允许访问。一旦 SRM 准许了对该对象的访问，以后就不需要再进行访问检查了，对象的句柄使得以后的访问可以直接进行。

对象的访问类型包括拒绝访问、读、修改、完全控制和特殊访问。目录的访问权限还包括列表、添加和读取。

Windows 使用句柄这个概念来访问系统中受保护的对象。每个对象拥有一个安全描述符(security descriptors，SD)，SD 中包含了对象的自主访问控制列表(discretionary access control list，DACL)。同时，每个进程拥有一个安全令牌，该令牌包含了一个用作识别进程的 SID。内核使用该 SID 来决定是否允许访问。ACL 中包含的 ACE 说明了允许某个进程 SID 对该对象进行哪些访问。内核遍历 ACL 来查找与请求的访问相对应的权限。

进程请求访问对象通常是采用请求对象的句柄方式，如 CreateFile 调用，CreateFile 调用既可以用来打开文件也可以创建一个新文件。当创建文件时，参数中包含一个指向 SD 的指针。当打开文件时，需要的参数包括文件句柄，以及所需的访问类型，如 GENERIC_READ。如果进程拥有相应访问类型所需的权限，则请求成功并且获得一个访问句柄。因此同一对象的不同句柄可能拥有不同类型的访问权限。一旦获得句柄，进程以后读文件时就不再需要更多的授权了。可以把句柄传递给其他可信的函数供后者使用。

Java 安全子系统也采用本节讨论的访问控制实施模型提供安全性。Java 的访问控制

器基于权限和策略构建访问权限。它使用一个 checkPermission 方法来决定每个调用方法的代码源对象,并使用当前的 Policy 对象确定与该方法绑定的权限对象。checkPermission 方法会遍历整个调用栈来决定栈中所有调用方法的访问。java.policy 文件中包含了每个代码源的授权声明,该文件供安全管理器使用。

采用上述访问控制模型实施访问控制,可以截取每个访问请求,并且根据授权规则来允许或拒绝该访问。访问规则可以实现一个访问矩阵,该矩阵定义了每个主体拥有的不同访问类型,可以按需要添加内容相关的规则。上述模型适用于实施各种访问控制策略,如 DAC、MAC 和 RBAC 等。

该模型的缺点是需要保护授权规则。同时,控制每个访问会带来一定的开销。对于那些内容相关的规则尤其严重,可以通过编译来提高效率。

8.2.3 访问控制实施策略

访问控制的实施策略包括入网访问控制策略、操作权限控制策略、目录安全控制策略、属性安全控制策略、网络服务器安全控制策略、网络监测和锁定控制策略及防火墙控制策略等。

1. 入网访问控制策略

入网访问控制是网络访问的第 1 层安全机制。它控制哪些用户能够登录到服务器并获准使用网络资源,控制准许用户入网的时间和位置。用户的入网访问控制通常分为三步执行:用户名的识别与验证;用户密码的识别与验证;用户账户的默认权限检查。三道控制关卡中只要任何一关未过,该用户便不能进入网络。

对网络用户的用户名和密码进行验证是防止非法访问的第一道关卡。用户登录时首先输入用户名和密码,服务器将验证所输入的用户名是否合法。用户的密码是用户入网的关键所在。密码最好是数字、字母和其他字符的组合,长度应不少于 6 个字符,必须经过加密。密码加密的方法很多,最常见的方法有基于单向函数的密码加密、基于测试模式的密码加密、基于公钥加密方案的密码加密、基于平方剩余的密码加密、基于多项式共享的密码加密和基于数字签名方案的密码加密等。经过各种方法加密的密码,即使是网络管理员也不能够得到。系统还可采用一次性用户密码,或使用如智能卡等便携式验证设施来验证用户的身份。

网络管理员应该可对用户账户的使用、用户访问网络的时间和方式进行控制和限制。用户名或用户账户是所有计算机系统中最基本的安全形式。用户账户应只有网络管理员才能建立。用户密码是用户访问网络所必须提交的准入证。用户应该可以修改自己的密码,网络管理员对密码的控制功能包括限制密码的最小长度、强制用户修改密码的时间间隔、密码的唯一性、密码过期失效后允许入网的宽限次数。针对用户登录时多次输入密码不正确

的情况,系统应按照非法用户入侵对待并给出警告信息,同时应该能够对允许用户输入密码的次数给予限制。

用户名和密码通过验证之后,系统需要进一步对用户账户的默认权限进行检查。网络应能控制用户登录入网的位置、限制用户登录入网的时间及限制用户入网的主机数量。当交费网络的用户登录时,如果系统发现“资费”用尽,还应能对用户的操作进行限制。

2. 操作权限控制策略

操作权限控制是针对可能出现的网络非法操作而采取的安全保护措施。用户和用户组被赋予一定的操作权限。网络管理员能够通过设置,指定用户和用户组可以访问网络中的哪些服务器和计算机,可以在服务器或计算机上操控哪些程序,访问哪些目录、子目录、文件和其他资源。网络管理员还应该可以根据访问权限将用户分为特殊用户、普通用户和审计用户,可以设定用户对可以访问的文件、目录、设备能够执行何种操作。特殊用户是指包括网络管理员的对网络、系统和应用软件服务有特权操作许可的用户;普通用户是指那些由网络管理员根据实际需要为其分配操作权限的用户;审计用户负责网络的安全控制与资源使用情况的审计。系统通常将操作权限控制策略,通过访问控制表来描述用户对网络资源的操作权限。

3. 目录安全控制策略

访问控制策略应该允许网络管理员控制用户对目录、文件和设备的操作。目录安全允许用户在目录一级的操作对目录中的所有文件和子目录都有效。用户还可进一步自行设置对目录下的子控制目录和文件的权限。对目录和文件的常规操作有读取(read)、写入(write)、创建(create)、删除(delete)和修改(modify)等。网络管理员应当为用户设置适当的操作权限,操作权限的有效组合可以让用户有效地完成工作,同时又能有效地控制用户对网络资源的访问。

4. 属性安全控制策略

访问控制策略还应该允许网络管理员在系统一级对文件、目录等指定访问属性。属性安全控制策略允许将设定的访问属性与网络服务器的文件、目录和网络设备联系起来。属性安全策略在操作权限安全策略的基础上,提供更进一步的网络安全保障。网络上的资源都应预先标出一组安全属性,用户对网络资源的操作权限对应一张访问控制表,属性安全控制级别高于用户操作权限设置级别。属性设置经常控制的权限包括向文件或目录写入、文件复制、目录或文件删除、查看目录或文件、执行文件、隐含文件、共享文件或目录等。允许网络管理员在系统一级控制文件或目录等的访问属性,可以保护网络系统中重要的目录和文件,维持系统对普通用户的控制权,防止用户对目录和文件的误删除等操作。

5. 网络服务器安全控制策略

网络系统允许在服务器控制台上执行一系列操作。用户通过控制台可以加载和卸载系统模块,可以安装和删除软件。网络服务器的安全控制包括可以设置密码锁定服务器控制台,以防止非法用户修改系统、删除重要信息或破坏数据。系统应该提供服务器登录限制、非法访问者检测等功能。

6. 网络监测和锁定控制策略

网络管理员应能够对网络实施监控。网络服务器应对用户访问网络资源的情况进行记录。对于非法的网络访问,服务器应以图形、文字或声音等形式告警,引起网络管理员的注意。对于不法分子试图进入网络的活动,网络服务器应能够自动记录这种活动的次数,当次数达到设定数值,该用户账户将被自动锁定。

7. 防火墙控制策略

防火墙是一种保护计算机网络安全的技术性措施,是用来阻止网络黑客进入企业内部网的屏障。防火墙分为专门设备构成的硬件防火墙和运行在服务器或计算机上的软件防火墙。无论哪一种,防火墙通常都安置在网络边界上,通过网络通信监控系统隔离内部网络和外部网络,以阻挡来自外部网络的入侵。

8. 访问控制管理

访问控制管理涉及访问控制在系统中的部署、测试、监控及对用户访问的终止。虽然不一定需要对每一个用户设定具体的访问权限,但是访问控制管理依然需要大量复杂和艰巨的工作。访问控制决定需要考虑机构的策略、员工的职务描述、信息的敏感性和用户的职务需求等因素。

有三种基本的访问管理模式:集中式、分布式和混合式。每种管理模式各有优缺点。应该根据机构的实际情况选择合适的管理模式。

(1)集中式管理

集中式管理就是由一个管理者设置访问控制。当用户对信息的需求发生变化时,只能由这个管理者改变用户的访问权限。由于只有极少数人有更改访问权限的权力,所以这种控制是比较严格的。每个用户的账号都可以被集中监控,当用户离开机构时,其所有的访问权限可以很容易地被终止。因为管理者较少,所以整个过程和执行标准的一致性就比较容易达到。但是,当需要快速而大量修改访问权限时,管理者的工作负担和压力就会很大。

(2) 分布式管理

分布式管理就是把访问的控制权交给了文件的拥有者或创建者，通常是职能部门的管理者(functional managers)。这就等于把控制权交给了对信息负有直接责任、对信息的使用最熟悉、最有资格判断谁需要信息的管理者的手中。但是这也同时造成在执行访问控制的过程和标准上的不一致性。在任意时刻，很难确定整个系统所有用户的访问控制情况。不同管理者在实施访问控制时的差异会造成控制的相互冲突，以致无法满足整个机构的需求。同时也有可能造成在员工调动和离职时访问权不能有效地清除。

(3) 混合式管理

混合式管理是集中式管理和分布式管理的结合。它的特点是由集中式管理负责整个机构中基本的访问控制，而由职能管理者就其所负责的资源对用户进行具体的访问控制。混合式管理的主要缺点是难以划分哪些访问控制应集中控制，哪些应在本地进行分布式控制。

8.2.4 访问控制语言

可扩展的访问控制标记语言(extensible access control markup language，XACML)是由 OASIS 组织开发，采用 XML 表示的访问控制策略语言。XACML 策略语言允许管理员定义访问控制需求。XACML 还包括一种访问决策语言，用于描述对资源运行时的请求。当确定了保护资源的策略之后，函数会将请求中的属性与包含在策略规则中的属性进行比较，最终生成一个许可或拒绝决策。简言之，XACML 是一种新的用于管理策略和访问控制的标记语言，同时又是一种通用的访问控制策略定义语言，提供一整套语法(使用 XML 定义)来管理对系统资源的访问。

目前，多数系统都以专有的方式实现访问控制和授权，在专有访问控制系统中，实体及其属性的信息保存在资料库，即访问控制列表中。不同的专有系统具有不同的实现 ACL 的机制，因此难以交换和共享信息。同时，这些机制缺少表示复杂策略(在现实系统中经常需要用到)的能力。因此，访问控制策略通常会嵌入应用程序代码中，这使得更改策略(或者只是找出哪些策略正在实施)变得很困难。

XACML 的出现使得不同环境中可以简单、灵活地制定各种访问控制策略。XACML 的通用性使得各系统之间的访问控制策略和过程得到标准化。XACML 是一种主要由机器生成的语言，它们能用于多个应用程序，并可以实现不同系统之间访问控制的互操作。

目前，XACML 的 API 已经由 Sun 公司实现，由 Java 语言写成，即 Sun's Java XACML Implementation。它是 Sun 提供的对于 XACML 标准的 Java 实现，是目前关于 XACML 开展最早的开源项目。Sun Java XACML 的实现实际上是一个对于 OASIS XACML 标准支持的类库，在 Sun Java XACML Implementation 1.2 中提供了对于 OASIS XACML 标准 2.0 版本的完全支持。

和其他专用访问控制策略语言比较，XACML 有如下优点。

① 安全管理员只需对访问控制策略描述一次，不必在不同的系统中使用不同的应用程序策略语言重写多次。

② 应用程序开发者不必开发自己的策略语言和编写支持它们的程序，他们可以重复使用已有的和标准化的程序。

③ XACML 能适应大多数访问控制策略的需求，当新的访问控制要求出现时，只需要加入策略，不必修改应用程序。

④ 单一 XACML 策略能应用于多个资源，这有助于在为不同资源编制策略时，避免不一致性和重复劳动。

⑤ XACML 中一个策略可以引用另一个策略。

如前所述，XACML 既是一种访问控制策略语言，也是访问控制判决请求/应答(request / response)语言。策略语言用于描述通用的访问控制需求，包括若干标准扩展点来定义新的函数、数据类型和组合逻辑等。请求/应答描述语言使得用户可以构成一个问询(query)来判断一个动作(action)是否被允许执行，并对结果进行解释。应答总是包含一个请求是否被允许的一个答案(answer)，它为下面 4 个值之一：Permit(允许)、Deny(拒绝)、Indeterminate(不确定、发生错误、无法做出判断)和 Not Applicable(不适用，该请求无法被服务器回应)。

XACML 的访问控制机制如图 8.35 所示。主体欲对某一资源(resource)执行某些动作时，访问请求被送往 PEP(policy enforcement point，策略执行点)，PEP 会根据请求者的属性、请求资源、动作及其他的附属信息来构成一个请求，然后 PEP 会将该请求发送到 PDP(policy decision point，策略判断点)。PDP 会查阅请求及一些关于该请求的访问控制策略信息，并最终回答该访问是否被承认合法，并将这个回答返回到 PEP，由 PEP 来对请求者做允许或拒绝的回答。PEP 和 PDP 可能被包容在同一个应用程序中，也可能被分布到不同的服务中。

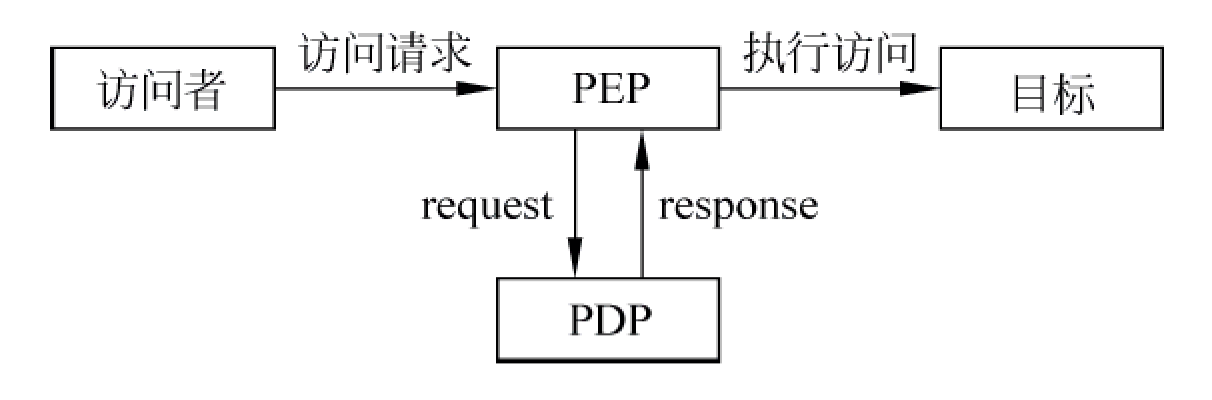


图 8.35　基于 XACML 的访问控制机制

8.2.5　安全审计

可以将安全审计定义为产生、记录并检查按时间顺序排列的系统事件记录的过程。安全审计利用计算技术，将计算机信息系统中发生的与安全相关的事件记录下来，并与安全规

则库进行比较，判断其是否违反了计算机系统的安全规定，或者提供事后追查的手段和方法来跟踪并处理安全问题。

通常，安全审计系统的业务主要包括如下几个部分。

① 硬件及环境审计：包括硬件安全、电源供应和空气调节等。

② 系统管理审计：包括对操作系统、数据库管理系统和所有系统过程审计。

③ 应用软件审计：包括访问控制、授权和确认等。

④ 网络安全审计：包括内部和外部的连接审计、周边安全、防火墙审计和端口扫描等。

⑤ 数据完整性审计：目的是详细检查有效数据来核实对系统弱点的控制。

⑥ 系统维护审计：包括容错及备份程序和存储，灾难恢复等。

1. 安全审计相关标准

审计作为一种安全保障机制，在最早的 TCSEC 中就已经有了明确的要求。历史上影响较大的两个安全评价标准 TCSEC 和 CC 都对审计提出了明确的功能要求。我国的国标《计算机信息系统安全保护等级划分准则》也有相应的规定。

(1) TCSEC 标准

TCSEC(trusted compute system evaluation criteria)俗称橙皮书，是美国国防部计算机安全中心于 1988 年发布的“可信计算机系统评估准则”，最初用于评估军用飞机的安全性和可靠程度。它从包括应用环境在内的各个角度出发，运用“级别确认”的方法对计算机系统进行安全评测。

从 C2 级开始要求具有审计功能，到 B3 级已经提出了关于审计的全部功能要求，A1 和 A2 两个级别较 B3 级没有增加任何安全审计特征。因此，TCSEC 共定义 T 的 4 个级别的审计要求：C2，B1，B2，B3。

C2 级要求审计以下事件：用户的身份标识和鉴别、用户地址空间中客体的引入和删除、计算机操作员/系统管理员/安全管理员的行为、其他与安全有关的事件。对于每一个审计事件，审计记录应包含以下信息：事件发生的日期和时间、事件的主体(即用户)、事件的类型、事件成功与否；对于用户鉴别这类事件，还要记录请求的来源(如终端号)；对于在用户地址空间中引入或删除客体，则要记录客体的名称；系统管理员对于系统内的用户和系统安全数据的修改也要在审计记录中得到体现。C2 级要求审计管理员应能够根据每个用户的身份进行审计。B1 级相对于 C2 级增加了以下需要审计的事件：对于可以输出到硬备份设备上的人工可读标志的修改(包括敏感标记的覆写和标记功能的关闭)、对任何具有单一安全标记的通信通道或 UO 设备的标记指定、对具有多个安全标记的通信通道或 UO 设备的安全标记范围的修改。因为增加了强制访问控制机制，B1 级要求在审计数据中也要记录客体的安全标记，同时审计管理员也可以根据客体的安全标记制定审计原则。B2 级的安全功能要求较之 B1 级增加了可信路径和隐蔽通道分析等，因此，除了 B1 级的审计要求外，对于可能被用于存储型隐蔽通道的活动，在 B2 级也要求被审计。B3 级在 B2 级的功能基础

上，增加了对可能将要违背系统安全政策这类事件的审计，比如对于时间型隐蔽通道的利用。审计子系统能够监视这类事件的发生或积聚，并在这种积聚达到某个阈值时立即向安全管理员发出通告。如果随后这类危险事件仍然持续下去，系统应在做出最小牺牲的条件下主动终止这些事件。这种及时通告意味着B3级的审计子系统不像其他较低的安全级别那样，只要求安全管理员在危险事件发生之后检查审计记录，而是能够更快地识别出这些违背系统安全政策的活动，并产生报告和进行主动响应。响应的方式包括锁闭发生此类事件的用户终端或者终止可疑的用户进程。一般的，“最小的牺牲”是与具体应用有关的，任何终止这类危险事件的行为都是可以接受的。

（2）TNI标准

TNI(trusted network interpretation)俗称“红皮书”，是美国国家计算机安全中心在1990年公布的“可信网络解释”准则，是将“橙皮书”中的评估原则应用于网络系统，并作为评价计算机网络安全保密特性的依据。它将整个网络的安全策略分解成若干策略单元，分别置于适当的、自主的网络组件中去实现，使其成为每一网络组件安全策略的基础，这样就从总体上确保了整个网络安全策略的有效实施。

（3）CC标准

CC(common criteria for information technology security evaluation)是国际标准化组织与国际电工委员会于1998年发表的《信息技术安全性评估通用准则2.0版》，即ISO/IEC 15408，称为CC标准。CC标准是信息技术安全性通用评估准则，用来评估信息系统或者信息产品的安全性。CC准则的安全功能需求定义了多个安全功能需求类，其中包括安全审计类。在CC准则中，对网络安全审计定义了一套完整的功能，如安全审计自动响应(security audit auto response)、安全审计事件生成(security audit data generation)、安全审计分析(security audit analysis)、安全审计浏览(security audit review)、安全审计事件选择(security audit event selection)和安全审计事件存储(security audit event storage)等。

（4）国标GB 17859-1999

依据我国计算机信息系统安全保护等级划分准则国标GB 17859-1999第三级中关于安全审计的要求，应确定如下的审计事件作为审计内容：身份鉴别机制的使用；将客体引入用户地址空间；客体的删除；操作员、系统管理员或系统安全管理员所实施的动作；其他的与安全相关的事件等。而每一事件的审计记录项应包括事件的日期与时间、用户、事件类型、实践成功与否。对于身份鉴别事件，审计记录还应包括请求的来源；对于客体引入用户地址空间的事件及客体删除事件，审计记录还应该包括客体的名称和客体的安全级别等。

2. 安全审计系统的主要功能

根据安全审计的相关标准和安全系统实际的需要，安全审计系统应具备如下功能。

（1）安全审计自动响应

安全审计自动响应是指当安全审计系统检测出一个安全违规事件(或者是潜在的违规)

时采取的自动响应措施。当检测到潜在的安全违规时，安全审计自动响应应该采取措施以避免即将来临的安全违规。例如安全告警，可以包括实施告警的生成、违例进程的终止、中断服务及用户账号的失效等。在实际应用中可以自定义多种响应措施，根据审计事件的不同，系统做出不同的响应。一个系统实现了安全审计自动响应将会实时通知管理员系统上发生的安全事件，某些自动响应措施还可以实时地降低损失。

(2) 安全审计数据生成

安全审计数据生成是指对安全功能控制下发生的安全相关事件进行记录，包括确定审计等级、列举可审计的事件类型，定义由不同类型审计记录提供的审计相关信息的最小集合。

产生的审计数据包括对于敏感数据项(例如密码等)的访问；目标对象的删除；访问权限或能力的授予和废除；改变主体或目标的安全属性；标识定义和用户授权认证功能的使用；审计功能的启动和关闭等。

安全审计数据生成功能应该在每条审计记录中至少记录以下信息。

① 事件发生的时间、事件类型、事件主标识及事件发生的结果(例如成功或失败)。

② 基于可审计事件功能组成的定义，对不同审计事件进行类型划分。

③ 用户相关标识，应该能够把每个可审计事件和产生此事件的用户标识关联起来。

(3) 安全审计分析

安全审计分析是指对系统行为和审计数据进行自动分析，以发现潜在的或实际发生的安全违规。安全审计分析的能力直接关系到能否识别真正的安全违规。安全审计分析需要入侵检测技术、自动响应技术的支持。安全审计分析包括潜在攻击分析、基于模板的异常检测、简单攻击试探和复杂攻击试探。

- 潜在攻击分析：系统能用一系列的规则监控审计事件，并根据规则识别系统的潜在攻击。
- 基于模板的异常检测：检测系统不同等级用户的行动记录，当用户的活动等级超过了限定的等级时，应指示其为一个潜在的攻击。
- 简单攻击试探：当发现一个系统事件与一个表示对系统潜在攻击的特征事件匹配时，应指示其为一个潜在的攻击。
- 复杂攻击试探：当发现一个系统事件或事迹序列与一个系统潜在攻击的特征事件匹配时，应指示其为一个潜在的攻击。复杂攻击试探法能够描绘和检测出多步骤的入侵攻击方案，能够对比系统事件(可能是多个个体实现)和事件序列来描绘出整个攻击方案，当发现某个特征事件或者事件序列时，能够表明发生了潜在的违规。

(4) 安全审计浏览

安全审计浏览是指经过授权的管理人员对于审计记录的访问和浏览。安全系统需要提供审计浏览的工具。通常审计系统对审计数据的浏览有授权控制，审计记录只能被授权的用户浏览，并且对于审计数据也是有选择地浏览。有些审计系统提供数据解释和条件搜索

等功能，帮助管理员方便地浏览审计记录。这里包括三种浏览方式：一般审计浏览、受限审计浏览和可选审计浏览。

- 一般审计浏览：提供从审计记录中读取信息的能力。应当做好对审计记录具有读访问权限的用户组的维护（删除、修改、添加）。提供授权用户得到审计记录信息，并且能够做出相应解释。
- 受限审计浏览：除了经过鉴定的授权用户，没有其他任何用户可以读取该信息。
- 可选审计浏览：可以通过审计工具按照一定标准来选择审计数据进行浏览。需要对审计数据提供逻辑关系上的查询、排序等能力。

（5）安全审计事件选择

安全审计事件选择是指管理员可以选择接受审计的事件，定义了从可审计的事件集合中选择接受审计的事件或者不接受审计的事件。一个系统通常不可能记录和分析所有的事件，因为选择过多的事件将无法实时处理和存储，所以安全审计事件选择的功能可以减少系统开销，提高审计的效率。此外，因为不同场合的需求不同，所以需要为特定场合配置特定的审计事件选择。安全审计系统应该能够维护、检查或修改审计事件的集合，能够选择对哪些安全属性进行审计，例如与目标标识、用户标识、主机标识或事件类型有关的属性。

（6）安全审计事件存储

安全审计事件存储主要是指对安全审计跟踪记录的建立、维护，如何保护审计，如何保证审计记录的有效性，以及如何防止审计数据的丢失等。审计系统需要对审计记录、审计数据进行严密保护，防止未授权修改。还需要考虑在极端情况下保护审计数据有效性，如存储介质失效、设计系统受到攻击等。

系统将提供控制措施以防止由于资源的不可用而丢失审计数据，能够创建、维护、访问它所保护的对象的审计踪迹，并保护其不被修改、非授权访问或破坏。审计数据将受到保护直至授权用户对它进行访问。它可保证某个指定量度的审计记录被维护，并不受以下事件的影响：审计存储空间用尽；审计存储故障；非法攻击；其他任何非预期事件。系统能在审计存储发生故障时或在审计存储即将用尽时采取相应的动作。

安全审计数据保护应包括如下方面。

① 受保护的审计跟踪存储。需要存储好审计跟踪记录，防止发生未授权删除或修改；

② 审计数据有效性的保证。需要保护存储的审计记录，防止发生未授权删除；需要能够防止或者检测出审计记录的修改；当存储介质异常、失效系统受到攻击时，应该保证审计记录的有效性。

③ 可能丢失数据情况下的措施。当审计记录数目超过预设值时，为了防止可能出现的审计数据丢失而需要采取一定措施。

④ 预防审计数据丢失。当审计记录跟踪用尽系统资源时，需要从以下措施中进行选择：忽略可审计事件；除了具有特殊权限的用户操作外，禁止可审计事件；覆盖旧的存储的审计记录等。

审计是系统安全策略的一个重要组成部分，它贯穿整个系统不同安全机制的实现过程，为其他安全策略的改进和完善提供了必要的信息。而且，它的深入研究为后来的一些安全策略的诞生和发展提供了契机。后来发展起来的入侵检测系统就是在审计机制的基础上得到启示而迅速发展起来的。

目前，网络安全审计系统刚刚起步不久，尚处在探索阶段，其审计重点也在网络的访问行为和网络中的各种数据。其中以 Purdue 大学的 NASHIS 系统较为著名。

安全审计系统作为一个完整安全框架中的一个必要环节，一般处在入侵检测系统之后，作为对防火墙系统和入侵检测系统的一个补充。因为网络安全审计系统能够分析出某些特殊的 IDS 无法检测的入侵行为（比如时间跨度很大的长期的攻击特征）；可以对入侵行为进行记录并能够在任何时间对其进行再现以达到取证的目的。同时，通过安全审计分析，可以提取一些未知的或者未被发现的入侵行为模式。

除了专门的安全审计系统外，操作系统或应用程序提供辅助手段进行安全相关事件的跟踪和审计。其中，日志是安全审计的一种重要手段，系统的日志记录提供了对系统活动的详细审计，这些日志用于评估、审查系统的运行环境和各种操作。一般日志记录包括记录用户登录时间、登录地点及进行什么操作等内容。日志记录可以向系统管理员提供有关危害安全的侵害或入侵试图等用于安全审计的有用信息。

8.3 防火墙技术

8.3.1 防火墙概述

如图 8.36 所示，防火墙是位于两个或多个网络之间，实施网间访问控制的一组组件的集合。在实施安全策略之后，防火墙能够限制被保护的内网与外网之间的信息存取和交换操作，作为不同网络或网络安全域之间交换信息的出入口，根据企业的安全策略控制出入网络的信息流。防火墙是一个分离器，一个限制器，也是一个分析器，控制内部网和外部网之间的活动，以提高内部网络的安全性。

一个部署了防火墙的系统应该具有如下特征。

① 内部网络和外部网络在物理上是连通的。

② 内部和外部之间的所有网络数据流必须经过防火墙。

③ 只有符合安全政策的数据流才能通过防火墙。

④ 防火墙自身应对渗透（penetration ）免疫。

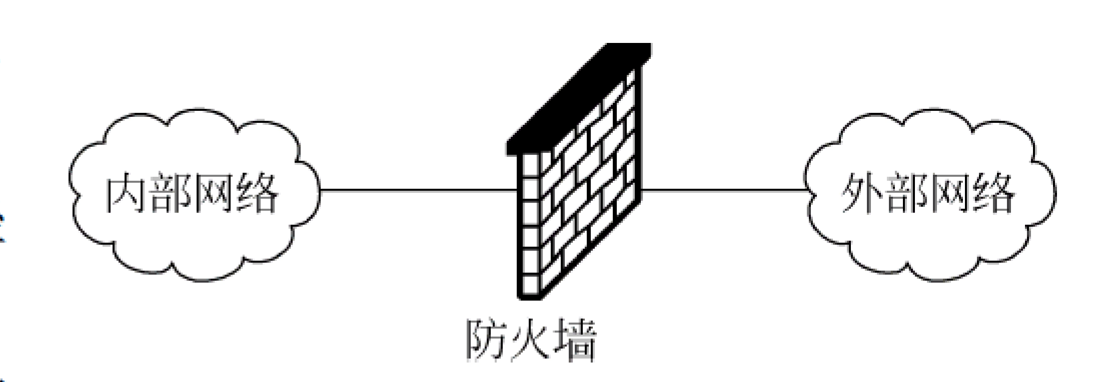


图 8.36 防火墙示意图

防火墙经历了一个发展过程，第一代防火墙采用包过滤技术。1989 年，推出了电路级

防火墙和应用层防火墙。1992 年,开发出了基于动态包过滤技术的防火墙。1998 年,NAI 公司推出了一种自适应代理技术。

设计和部署防火墙时须考虑两个因素,即安全性和实用性。应该综合考虑安全性和实用性,在满足安全性的前提下尽量提高其可用性,把它对系统的影响降低。目前,防火墙设计时多数采用的是“没有被允许就是禁止”的策略,就是将防火墙的安全性放在首位,这也正是设计者的初衷。同时,在足够安全的情况下,应增加防火墙的实用性。

目前防火墙在应用上仍存在一些不足,例如下面这些方面。

① 不能防止内部的人为破坏。一旦进入到系统内部或取得了系统控制权,那么防火墙就形同虚设了。也就是说,防火墙不能保护发生在子网(或节点)内部的破坏。

② 目前的防火墙还不能有效防御新的特洛伊木马等恶意程序,它只是一种被动式的防御手段。另外,对于来自非网络而是存储介质上的恶意程序,也不能起到阻挡作用。同时,防火墙很难防范数据驱动型的攻击。数据驱动型的攻击从表面上看是无害的,数据被邮寄或复制到 Internet 主机上,实际上,数据包中包括了一些隐藏的指令,一旦执行就开始攻击,从而使入侵者获得对系统的访问权。

③ 防火墙为了提高被保护网络和节点的安全性,可能会限制或关闭一些有用但存在安全缺陷的网络服务,从而损失可用性。由于绝大多数网络服务设计之初没有考虑安全性,只考虑使用的方便性和资源共享,所以安全问题普遍存在。如果完全禁止这些服务,则会损失可用性。

④ 防火墙会在一定程度上影响网络性能。例如,布置了防火墙后,通过它再访问 Internet 时带宽就会被降低。

⑤ 可移植性问题。到目前为止,尚不存在真正开放的、与软硬件平台无关的防火墙软件产品。

⑥ 多数防火墙不具有专业防御病毒的功能,必须使用防病毒软件来防御病毒。

8.3.2 防火墙分类

防火墙中可使用多种网络安全技术和手段,但总体来讲,可以将防火墙分为“包过滤型”和“应用代理型”两大类。

1. 包过滤(packet filtering)型防火墙

如图 8.37 所示,包过滤型防火墙工作在 OSI 网络参考模型的网络层和传输层,它根据数据包的源地址、目的地址、端口号和协议类型等标志确定是否允许数据包通过。只有满足过滤条件的数据包才被转发到相应的目的地,其余数据包则被从数据流中丢弃。

包过滤方式是一种通用、廉价和有效的安全手段。之所以通用,是因为它不是针对各个具体的网络服务采取特殊的处理方式,适用于所有网络服务;之所以廉价,是因为大多数路

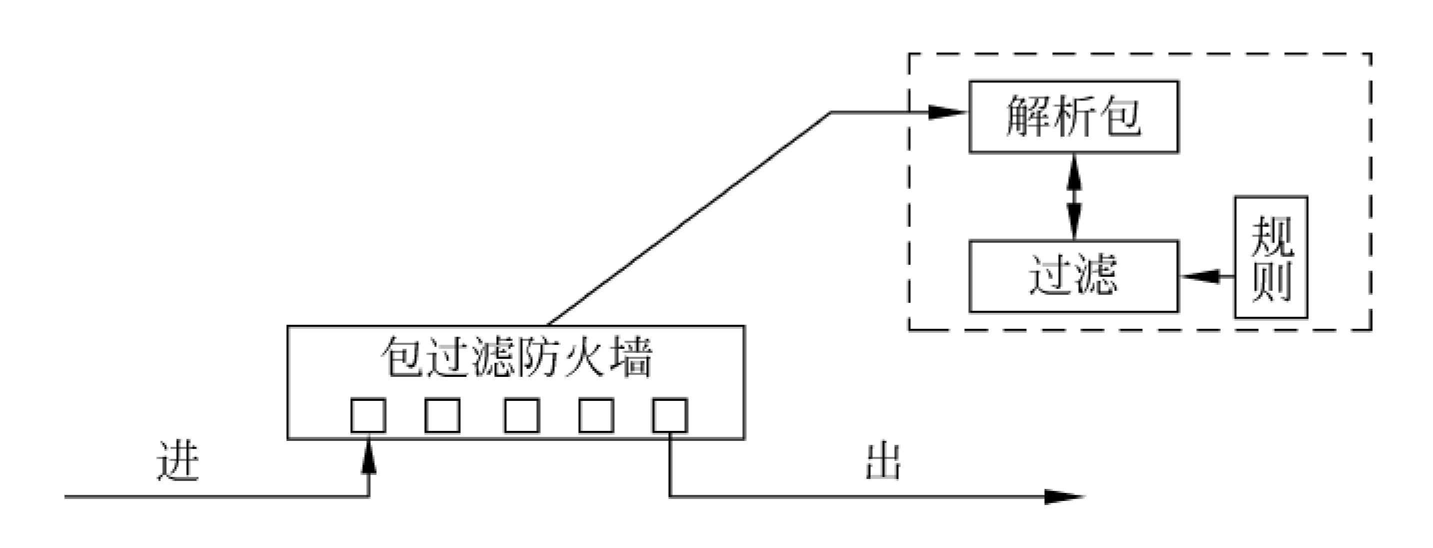


图 8.37 包过滤防火墙示意图

由器都提供数据包过滤功能，所以这类防火墙多数是由路由器集成的；之所以有效，是因为它能在很大程度上满足绝大多数企业的安全要求。

包过滤技术的具体实现和应用包括商业版防火墙产品、个人防火墙、路由器和开源软件(如 Ipfilter 、Ipchains 和 Iptables)等。

在防火墙技术的发展过程中，包过滤技术出现了两种不同版本，称为“静态包过滤”和“动态包过滤”。

(1) 静态包过滤防火墙

这类防火墙几乎是与路由器同时产生的，它根据定义好的过滤规则审查每个数据包，以便确定其是否与某一条包过滤规则匹配。过滤规则基于数据包的报头信息进行制定。报头信息中包括 IP 源地址、IP 目标地址、传输协议(如 TCP、UDP 和 ICMP 等)、TCP/UDP 目标端口和 ICMP 消息类型等。

静态包过滤防火墙的协议层次如图 8.38 所示。

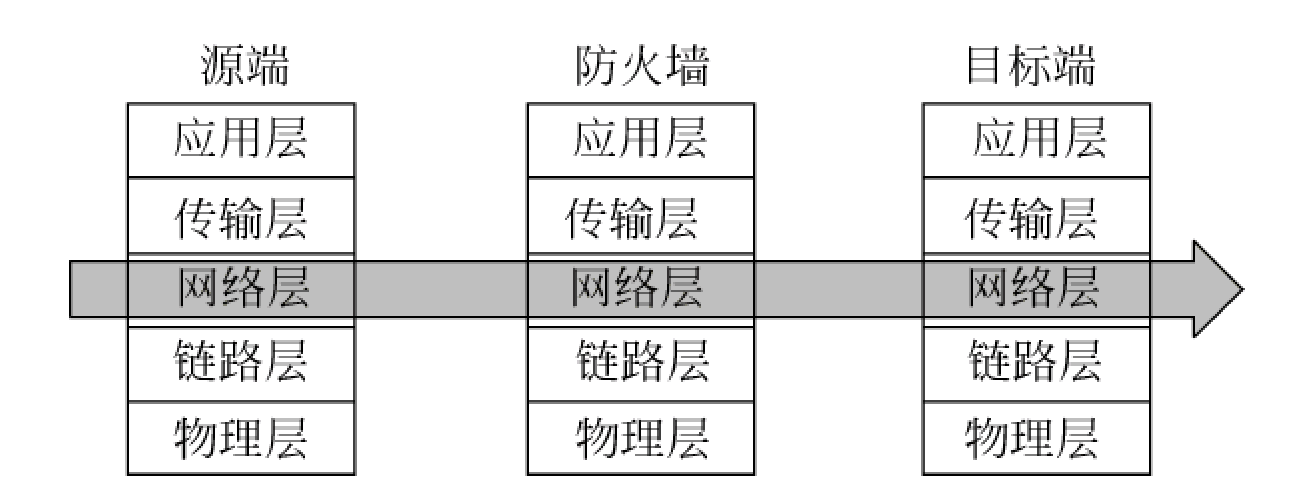


图 8.38 静态包过滤防火墙的协议层次

(2) 动态包过滤防火墙

这类防火墙采用动态设置包过滤规则的方法，这种技术后来发展成为包状态监测(stateful inspection)技术。采用这种技术的防火墙对通过其建立的每一个连接都进行跟踪，并且根据需要可动态地在过滤规则中增加或更新条目。

包过滤方式的优点是不用改动客户机和主机上的应用程序，因为它工作在网络层和传输层，与应用层无关。但其弱点也是明显的：过滤判别的依据只是网络层和传输层的有限信息，因而各种安全要求不可能充分满足；在许多过滤器中，过滤规则的数目是有限制的，且随着规则数目的增加，性能会受到很大影响；由于缺少上下文关联信息，不能有效地过滤

如 UDP、RPC(远程过程调用)一类的协议。另外,大多数过滤器中缺少审计和告警机制,它只能依据包头信息,而不能对用户身份进行验证,很容易受到“地址欺骗型”攻击。对安全管理人员素质要求高,建立安全规则时,必须对协议本身及其在不同应用程序中的作用有较深入的理解。因此,过滤器通常和应用网关配合使用,共同组成防火墙系统。

动态包过滤防火墙的协议层次如图 8.39 所示。

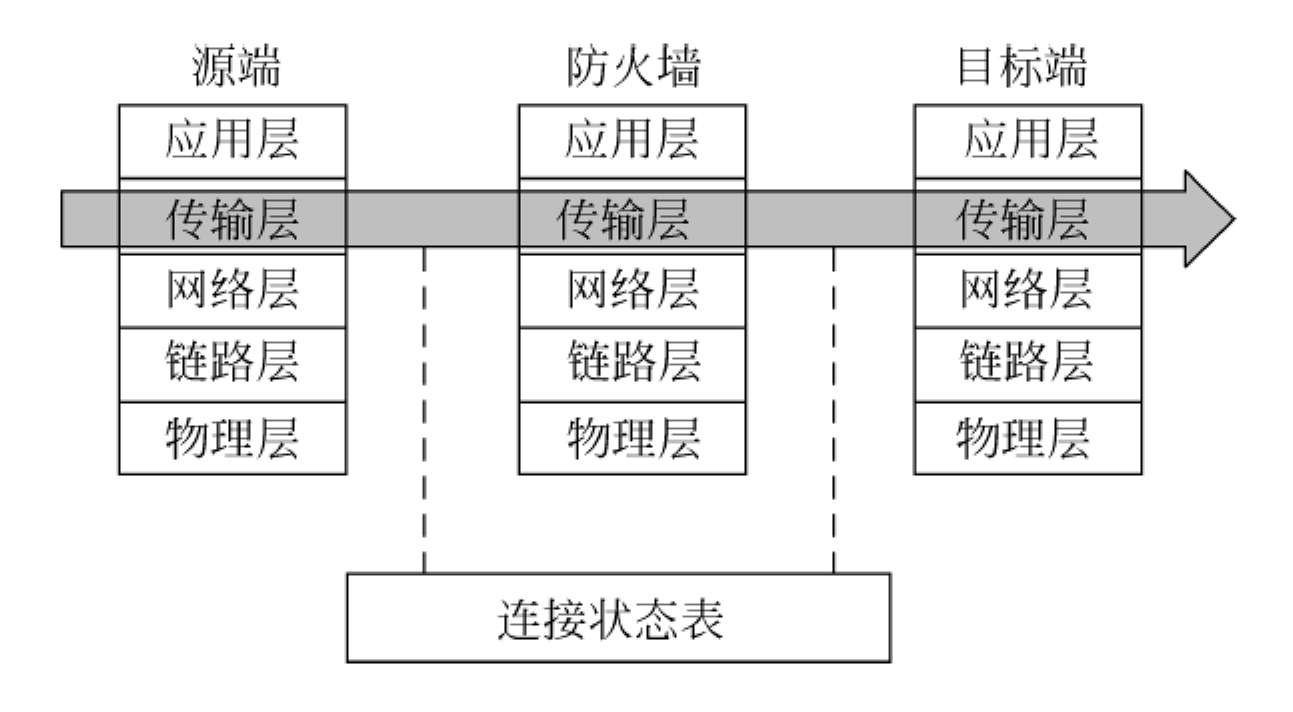


图 8.39　动态包过滤防火墙的协议层次

2. 代理(application proxy)型防火墙

代理防火墙工作在 OSI 的高层,即应用层或传输层与应用层之间(如 socks 服务器)。其特点是完全“阻隔”了网络通信流,通过对每种应用服务编制专门的代理程序,实现监视和控制应用层通信流的作用。其典型网络结构如图 8.40 所示。

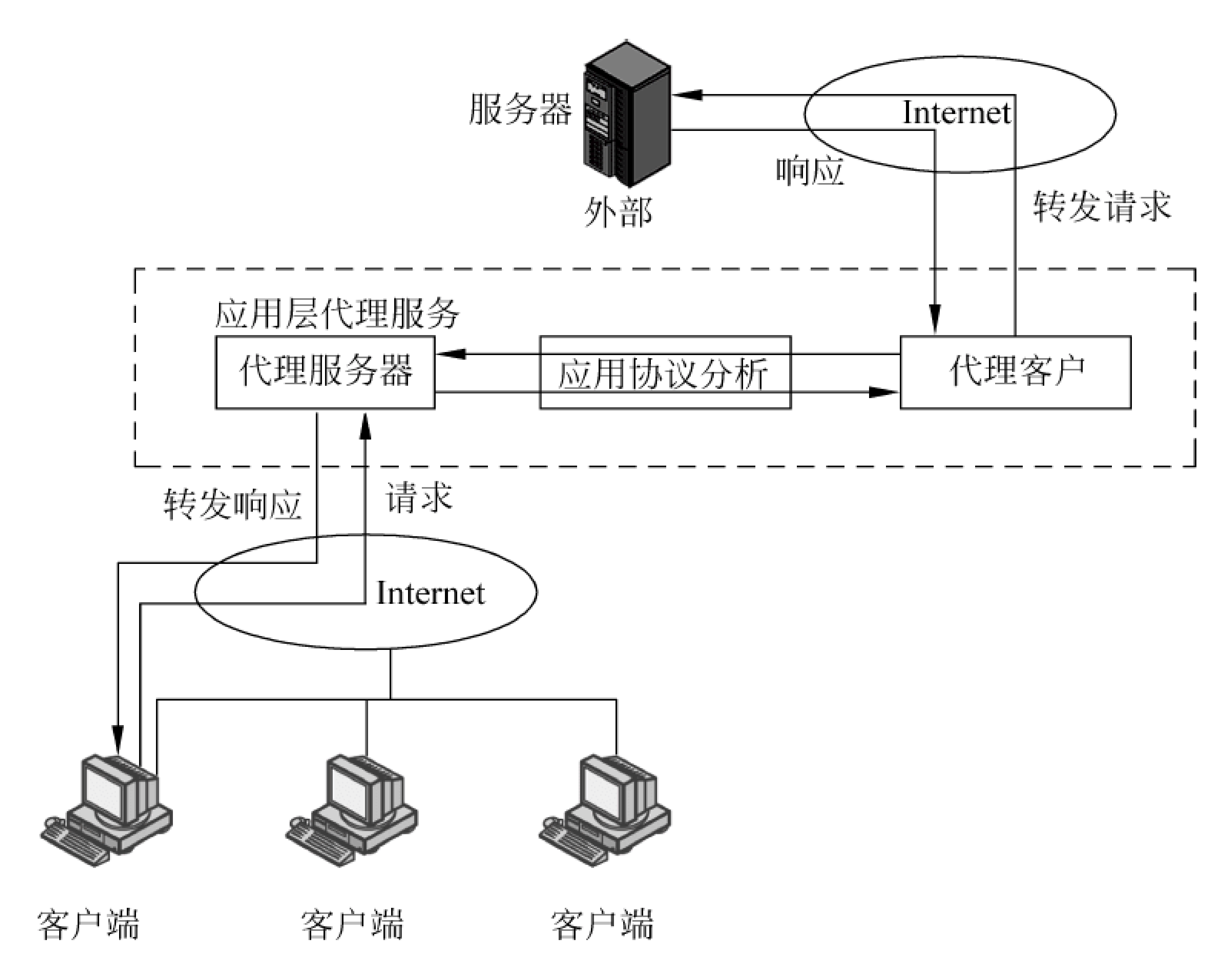


图 8.40　代理型防火墙的工作示意图

代理型防火墙的实现及其应用包括各类代理型服务器，如 wingate、ccproxy 和 sygate 等。

进一步可以把代理型防火墙分为应用网关型防火墙、电路级代理型防火墙（电路级网关）和自适应代理防火墙。

（1）应用网关（application gateway）

这类防火墙是通过一种代理（proxy）技术参与到一个 TCP 连接的全过程。从内部发出的数据包经过这样的防火墙处理后，就好像是源于防火墙外部网卡一样，从而可以达到隐藏内部网结构的作用。这种类型的防火墙被网络安全专家和媒体公认为是最安全的防火墙。它的核心技术就是代理服务器技术。其工作的协议层次如图 8.41 所示。

图 8.41　应用网关防火墙的协议层次

（2）电路级网关（circuit gateway）

电路级网关的工作原理和应用网关型防火墙相同，它接收客户端连接请求，代理客户端完成网络连接，在客户和服务器之间中转数据。和应用程序网关不同的是，它不是针对不同的应用程序设计的，而是工作在应用层和传输层之间（其协议层次如图 8.42 所示）。它根据客户的地址及所请求端口，将该连接重定向到指定的服务器地址及端口上，对客户端应用完全透明，通用性强。实现电路级网关的典型协议是 socks（见 6.3 节）。

图 8.42　电路级网关的协议层次

目前，许多商业代理服务器均支持应用层代理服务器和电路级网关。

（3）自适应代理（adaptive proxy）防火墙

它是近几年才得到广泛应用的一种新防火墙类型。它可以结合代理类型防火墙的安全性和包过滤防火墙的高速度等优点，在毫不损失安全性的基础之上将代理型防火墙的性能提高 10 倍以上。组成这种类型防火墙的基本要素有两个：自适应代理服务器（adaptive proxy server）与动态包过滤器（dynamic packet filter）。

在"自适应代理服务器"与"动态包过滤器"之间存在一个控制通道。在对防火墙进行配置时，用户仅仅将所需要的服务类型、安全级别等信息通过相应 Proxy 的管理界面进行设置就可以了。然后，自适应代理就可以根据用户的配置信息，决定是使用代理服务从应用层代理请求还是从网络层转发包。如果是后者，它将动态地通知包过滤器增减过滤规则，满足用户对速度和安全性的双重要求。

代理型防火墙最突出的优点是安全。由于它工作于高层协议，所以可以对网络中任何一层数据通信进行筛选保护，而不是像包过滤那样，只是对网络层的数据进行过滤。

此外，代理型防火墙采取的是一种代理机制，可以为每一种应用服务建立一个专门的代理。所以内外部网络之间的通信不是直接的，而都需先经过代理服务器审核，通过后再由代理服务器代为连接，根本没有给内、外部网络计算机任何直接会话的机会，从而避免了入侵者使用数据驱动类型的攻击方式入侵内部网。

代理型防火墙的缺点是速度相对比较慢，当用户对内外部网络网关的吞吐量要求比较高时，代理型防火墙就会成为内外部网络之间的瓶颈。那是因为防火墙需要为不同的网络服务建立专门的代理服务，在自己的代理程序为内、外部网络用户建立连接时需要时间，所以给系统性能带来了一些负面影响，但通常不会很明显。

包过滤和代理防火墙有各自的特点和适用场合，表 8.3 中给出了两者之间的优缺点比较。

表 8.3　包过滤和代理防火墙的特点比较

	包过滤防火墙	代理防火墙
优点	价格较低	内置了专门为了提高安全性而编制的 Proxy 应用程序，能够透彻地理解相关服务的命令，对来往的数据包进行安全化处理
	性能开销小，处理速度较快	安全，不允许数据包通过防火墙，避免了数据驱动式攻击的发生
缺点	定义复杂，容易出现因配置不当带来的问题	速度较慢，不太适用于高速网(ATM 或千兆位 Intranet 等)之间的应用
	允许数据包直接通过，容易造成数据驱动式攻击的潜在危险	
	不能理解特定服务的上下文环境，相应控制只能在高层由代理服务和应用层网关来完成	

8.3.3　防火墙相关技术

防火墙使用的相关技术主要有包过滤技术、代理技术、地址翻译技术和 VPN 技术(见 8.1 节)等。

1. 包过滤技术

包过滤技术是防火墙中的一项主要安全技术，它通过防火墙对进出网络的数据流进行控制与操作。系统管理员可以设定一系列规则，允许指定哪些类型的数据包可以流入或流出内部网络；哪些类型的数据包传输应该被拦截。现在的一些包过滤防火墙不仅根据IP数据包的地址、方向、协议、服务、端口和访问时间等信息来进行访问控制，同时还对任何网络连接和当前的会话状态进行分析和监控。

一般，包过滤的判断依据如下。

- 数据包协议类型：如TCP、UDP、ICMP和IGMP等。
- 源、目的IP地址。
- 源、目的端口：如FTP、HTTP和DNS等。
- IP选项：源路由、记录路由等。
- TCP选项：如SYN、ACK、FIN和RST等。
- 其他协议选项：如ICMP ECHO、ICMP ECHO REPLY等。
- 数据包流向：in或out。
- 数据包流经网络接口：eth0、eth1。

包过滤防火墙可以安装在一个路由器上（或是一个双端口网关上，也可以安装在一台服务器上）。由于Internet与Intranet的连接多数都要使用路由器，所以Router成为内外通信的必经端口，路由器的厂商在路由器上加入IP包过滤功能。这种防火墙在合理配置的前提下可以提供一定的安全性，然而一个包过滤规则是否完全严密及必要是很难判定的，因而在安全要求较高的场合，通常还配合使用其他的技术来加强系统的安全性。

包过滤防火墙一般有一个包检查模块，数据包过滤可以根据IP数据包头中的各项信息来控制站点与站点、站点与网络、网络与网络之间的相互访问。但是，包过滤防火墙不能控制传输数据的内容，因为内容是应用层数据，不是包过滤系统所能辨认的。

包过滤模块一般会在操作系统或路由器转发包之前拦截所有的数据包。验证这个包是否符合过滤规则，并记录数据包的情况（如数据包到达或离开的接口）。对符合规则的包进行处理转发，对不符合的包进行告警或通知管理员。如果无匹配规则，那么，用户配置的默认参数将决定此包是前行还是被舍弃。过滤规则指的是系统管理员依据本部门的安全策略起草的规则集。

传统静态包过滤型防火墙的包过滤只是与规则表进行匹配，对符合规则的数据包进行处理，不符合规则的丢弃。由于是基于规则的检查，属于同一连接的不同包毫无任何联系，每个包都要依据规则顺序过滤，这样随着安全规则的增加，势必会使防火墙的性能大幅度地降低，造成网络拥塞。甚至黑客会采用IP Spoofing（IP欺骗）的办法将自己的非法包伪装成属于某个合法的连接。这样的包过滤既缺乏效率又容易产生安全漏洞。

动态包过滤防火墙采用了基于连接状态检查的包过滤，将属于同一连接的所有数据包

作为一个整体的数据流看待，通过规则表与连接状态表的共同配合，大大地提高了系统的性能和安全性。现在的一些包过滤防火墙在进行包的检测时不仅将其看成是独立的单元，同时还要考虑与它的前面包的关联性。

与静态包过滤技术不同，动态包过滤防火墙知道一个新的连接和一个已经建立的连接的不同。对于已经建立的连接，动态包过滤防火墙将状态信息写进常驻内存的状态表，后来包的信息与状态表中的信息进行比较，该动作是在操作系统的内核中完成的，因此增加了安全性。一个典型的例子是，静态包过滤无法区分一个外部用户进入的包与一个内部用户出去后回来的包的不同，动态包过滤防火墙就知道。动态包过滤防火墙可以限制外部用户访问内部，但保证内部用户可以访问外部，而且可以回来。

当一个包是属于一个已经建立连接时，防火墙不作进一步检查就可以放行这个包。通过占有部分系统内存，减少了包的检查工作量，因此，动态包过滤的性能有一定程度的增加。

和静态包过滤机制比较，动态包过滤技术可以支持对称多处理系统（symmetrical muti-processing，SMP）和多 CPU 系统，可以取得更高的速度和性能。例如，在基于 TCP 协议的连接中，每个包在传输时都包括了 IP 源地址、目的地址、源端口和目的端口等信息，还包括了对在允许的时间间隔内是否发生了 TCP 握手消息的监视信息等。这些信息与每个数据包都是有关联的。换句话说，对于属于同一个连接的数据包来说并不是完全孤立的，它们存在内部的关联信息。无连接的包过滤规则没有考虑这些内在的关联信息，而是对每个数据包都进行孤立的规则检测，这样就降低了传输效率和安全性。

值得一提的是，对于基于 UDP、ICMP 协议的应用来说，很难用简单的包过滤技术进行处理，因为 UDP 协议本身对于顺序错误或丢失的包，不做纠错或重传。而 ICMP 与 IP 位于同一层，它被用来传送 IP 的差错和控制信息。现在的一些包过滤防火墙在对基于 UDP 协议的连接处理时，会为 UDP 建立虚拟的连接，同样能够对连接过程状态进行监控。通过规则与连接状态的共同配合，达到包过滤的高效与安全。因此能够实现对 UDP、ICMP 协议的实时状态监控。这种实时的动态过滤技术使防火墙增强了防御能力，降低了黑客攻击的成功率，从而提高了系统的性能和安全性。另外，现在包过滤技术逐渐从一种被动的规则检查方式变为多级并行或串行，或者串并行混合的复杂检查方式，例如入侵检测与防火墙互动。

对于一个大型网络，定义包过滤器规则是一项复杂的工作，因为网管员需要详细地了解 Internet 的各种服务、包头格式和他们在希望每个域查找的特定的值。如果必须支持复杂的过滤要求，则过滤规则集可能会变得很长和很复杂，从而很难管理。存在几种自动测试软件，被配置到 Router 上后即可校验过滤规则。一般来说，吞吐量随过滤器数量的增加而减少。如果包过滤须对每个包执行所有过滤规则的话，这可能消耗 CPU 的资源，并影响一个完全饱和的系统性能。

下面给出一个包过滤的示例，网络拓扑如图 8.43 所示。

在图 8.43 所示网络中，内部网地址为 192.168.0.0/24，堡垒主机中与内部网络相连的

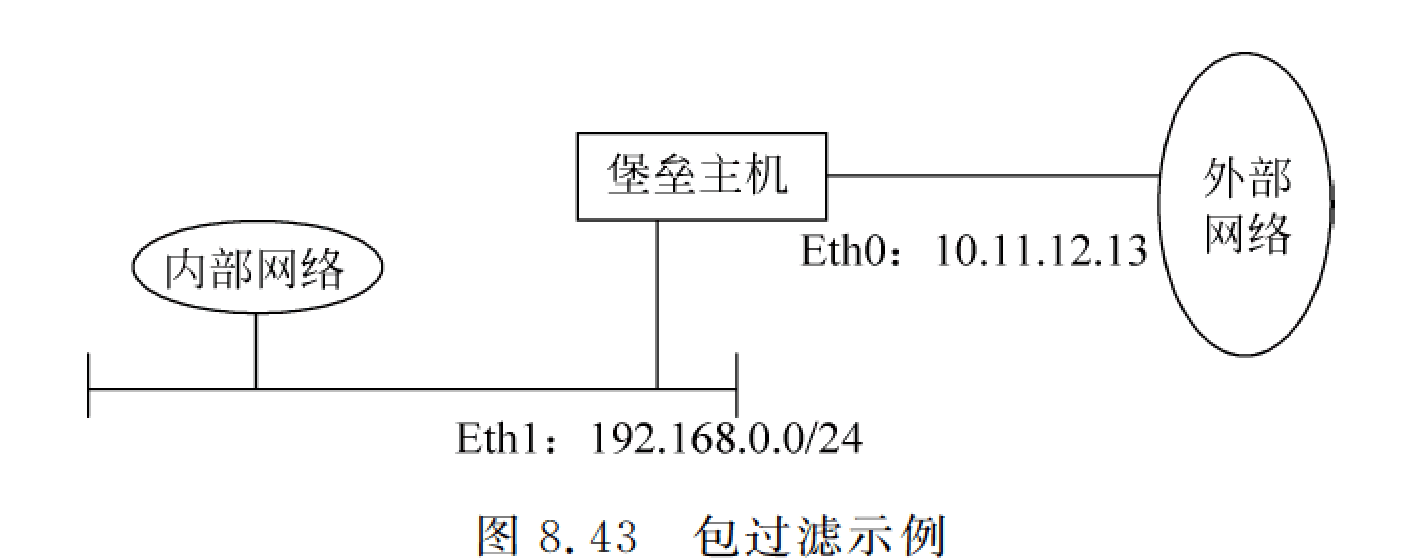


图 8.43 包过滤示例

网卡接口为 Eth1，地址为 192.168.0.1，外网相连的网卡接口为 Eth0，地址为 10.11.12.13，DNS 服务器的地址为 10.11.15.4。其中，堡垒主机为安装了多个网络接口并配置了路由器功能的计算机，在系统中承担防火墙的功能。假设系统要求允许内网所有主机能访问外网的 WWW 和 FTP 服务，外网不能访问内部主机，则堡垒主机上的包过滤规则配置如下。

```
Set internal = 192.168.0.0/24
Deny ip from $internal to any in via eth0
Deny ip from not $internal to any in via eth1
Allow udp from $internal to any dns
Allow udp from any dns to $internal
Allow tcp from any to any established
Allow tcp from $internal to any www in via eth1
Allow tcp from $internal to any ftp in via eth1
Allow tcp from any ftp-data to $internal in via eth0
```

2. 代理技术

如 8.3.2 节中所述，代理技术指的是应用代理或代理服务器技术，可具体为一个代理内部网络用户与外部网络服务器进行信息交换的程序。它将内部用户的请求确认后送达外部服务器，同时将外部服务器的响应再回送给用户。

代理服务器又称应用程序网关，它可以屏蔽内部网的细节，使非法用户无法探知内部网络的结构。它能够屏蔽某些特殊的命令，禁止用户使用容易造成攻击的不安全的命令，从而抵御攻击。同时，代理服务器还能够过滤非安全脚本，如 ActiveX、Java Applet、Java Script 及进行邮件过滤。

现在的一些包过滤防火墙中对 FTP、TELNET、HTTP、SMTP、POP3 和 DNS 等应用实现了代理服务。这些代理服务对用户是透明的，即用户意识不到防火墙的存在便可完成内外网络的通信。当内部用户需要使用透明代理访问外部资源时，用户不需要进行设置，代理服务器会建立透明的通道，让用户直接与外界通信，这样极大地方便了用户的使用，避免使用中的错误，降低使用防火墙时固有的安全风险和出错概率。

现在的一些包过滤防火墙的代理服务器提供了对连接流量的控制功能，系统管理员可以根据内部网络的需要增大或减少某一代理 FTP、HTTP、TELNET、SMTP、POP3 和 DNS

等的流量,这样能更有效地利用资源,也减轻了防火墙的负荷。并且,现在的一些包过滤防火墙采用了多线程多连接技术,使系统可以对出入防火墙的所有应用层连接进行统一的管理,处理速度快,处理进程多,保证了系统的高效性。

3. 网络地址转换技术

网络地址转换(network address translation,NAT)可以对外部网络隐藏内部的网络结构,使得外部攻击者无法确定内部计算机的连接状态。NAT 功能通常被集成到路由器、防火墙中。NAT 设备维护一个状态表,用来把内部私有 IP 地址映射到因特网的公共地址上。每个 IP 包在 NAT 设备中都被翻译成因特网 IP 地址发往下一级,该数据包像正常的数据包一样在网络上进行转发,直到目的地。当 IP 数据包返回时,经过 NAT 设备再次进行地址转换,将公网地址翻译为最初的私有地址转发给正确的内部主机。

如图 8.44 所示,内部主机 192.168.0.2 使用其私有地址访问因特网,在防火墙上启动了 NAT 功能后,该内部私有地址被防火墙替换为 202.112.136.196,以这个公网地址作为源地址的 IP 包被转发到目标地址,从目标地址回来的数据包再次被 NAT 设备(防火墙)翻译为 192.168.0.2 转发给该内部主机。

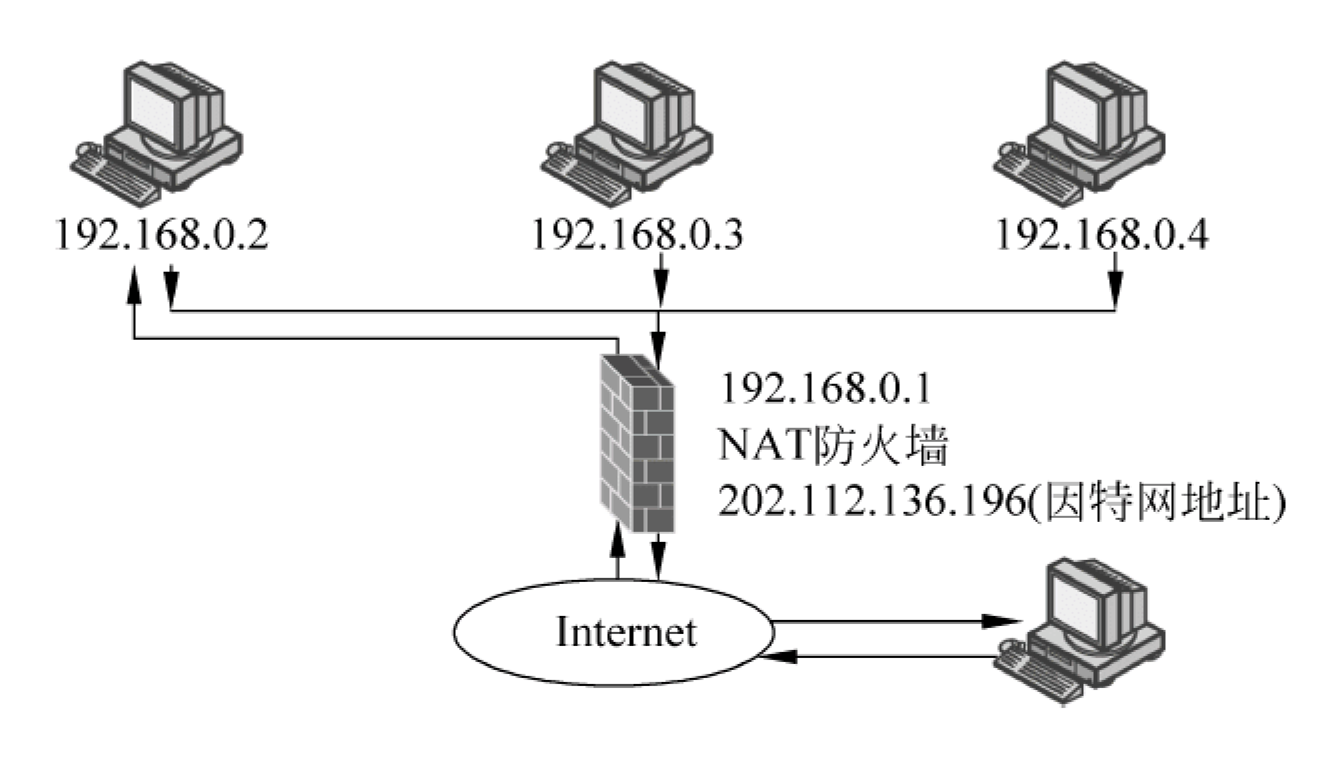


图 8.44　使用 NAT 的网络示例

在图 8.44 中,网络地址转换的详细工作过程如图 8.45 所示,其地址转换过程描述如下。

① 内部主机使用内部地址 192.168.0.2 向因特网上的主机 128.119.40.186 的 80 号端口发送数据包,该数据包首先发往主机的默认网关 192.168.0.1。

② 默认网关上事先配置了 NAT,指明内部地址为 192.168.0.2 的主机使用 202.112.136.196 这个因特网合法地址访问因特网。网关接收来自 192.168.0.2 的数据包后,对该数据包进行地址转换,把源地址和端口号替换为 202.112.136.196:5001。同时,在 NAT 转换表中为该数据包建立一条 NAT 记录,其中指明了内部地址、端口号及对应的外部地址(因特网地址)及端口号,此例中分别为 192.168.0.2:2345 及 202.112.136.196:5001。随后,该数据包被转发到因特网上。

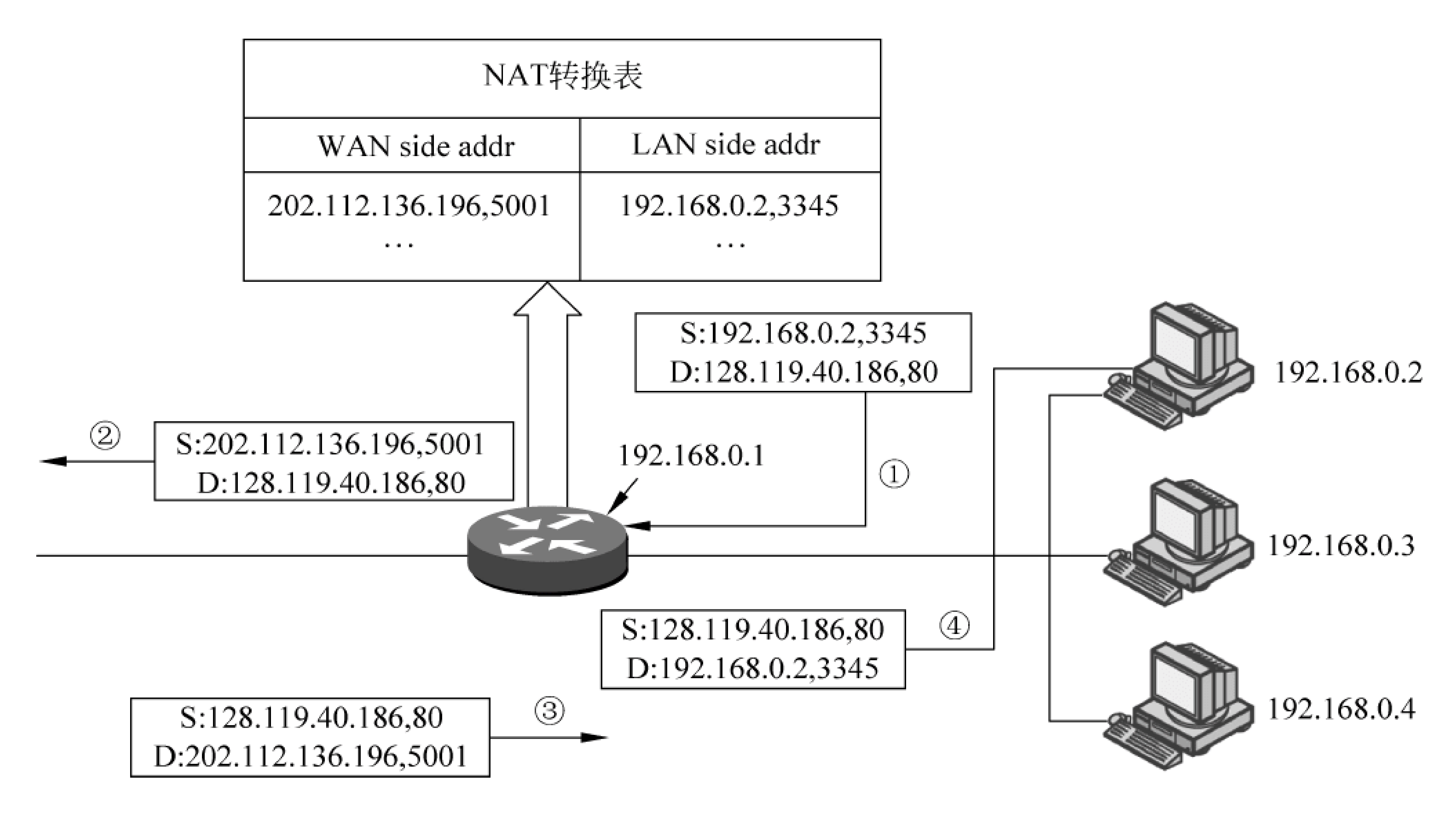


图 8.45　NAT 的工作原理及工作过程示意图

③ 因特网上的目标主机 128.119.40.186 对请求数据包进行应答，应答数据包的目标地址为 202.112.136.196。

④ 网关根据②中的 NAT 记录，将应答数据包的目标地址和端口号替换为最初的值，即 192.168.0.2：2345。该数据包被转发到内部主机上。

4. 其他相关技术

① 加密技术。网络上传输信息的私有性和完整性可以用加密技术解决。在应用中，它应该包括三个部分：加密算法的选择，信息确认算法的选择及产生和分配密钥的密钥管理协议。

② 安全审计。绝对的安全是不可能的，因此必须对网络上发生的事件进行记载和分析，对某些被保护网络的敏感信息访问保持不间断的记录，并通过各种不同类型的报表、告警等方式向系统管理人员进行报告。比如在防火墙的控制台上实时显示与安全有关的信息、对用户密码非法或非法访问进行动态跟踪等。

③ 安全内核。除了代理服务器以外，人们开始在操作系统的层次上考虑安全性。例如考虑把系统内核中可能引起安全问题的部分从内核中去除，形成一个安全等级更高的内核，从而使系统更安全，如 Cisco 的 PIX 防火墙等。

④ 身份认证。目前，一般防火墙主要提供如下认证方法。

- 用户认证（user authentication，UA）：防火墙设定可以访问内部网络资源的用户访问权限。
- 客户认证（client authentication，CA）：防火墙提供特定用户端授权用户特定的服务权限。

- 会话认证(session authentication,SA):防火墙提供通信双方每次通信时透明的会话授权机制。

⑤ 负载平衡(overload balance)。平衡服务器的负载,由多个服务器为外部网络用户提供相同的应用服务。当外部网络的一个服务请求到达防火墙时,防火墙可以用其制定的平衡算法确定请求由哪台服务器来完成。但对用户来讲,这些都是透明的。

8.3.4 防火墙应用模式

1. 多穴主机模式

多穴主机(multihomed host)防火墙,顾名思义,它用一台拥有两块或两块以上网卡的主机(该主机称为堡垒主机)作为防火墙,网卡分别连接到物理和逻辑上都分离的不同网段上,从逻辑上使一个子网与其他子网、内部网与外部网之间不能进行直接通信,从而起到保护自身不受入侵的目的。

使用双穴和多穴主机的防火墙系统的示意图分别如图 8.46 和图 8.47 所示。

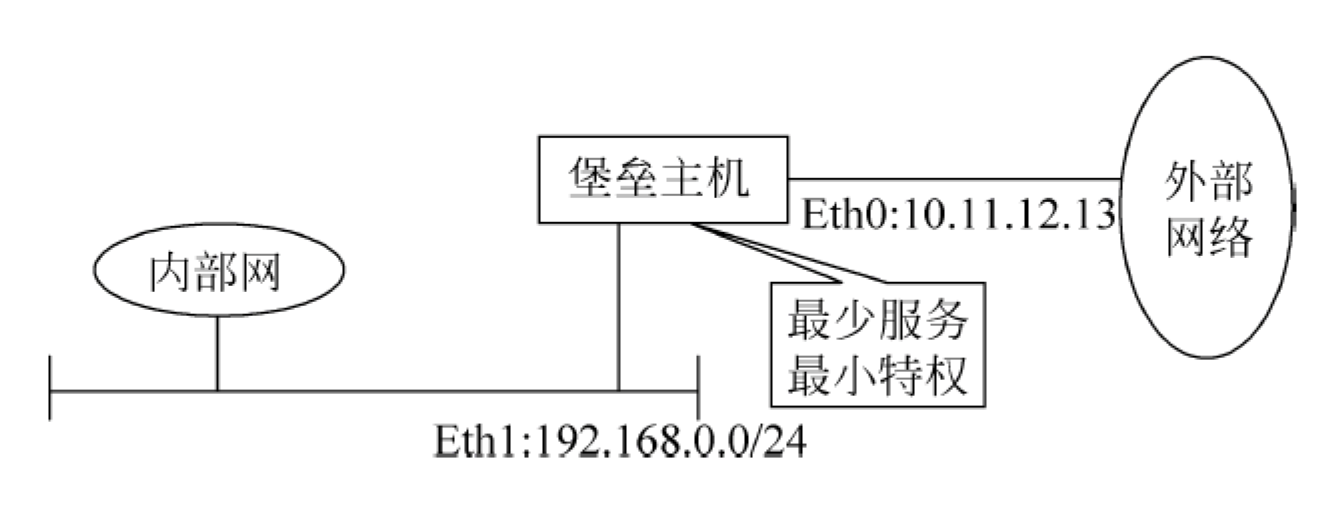


图 8.46 双穴主机模式

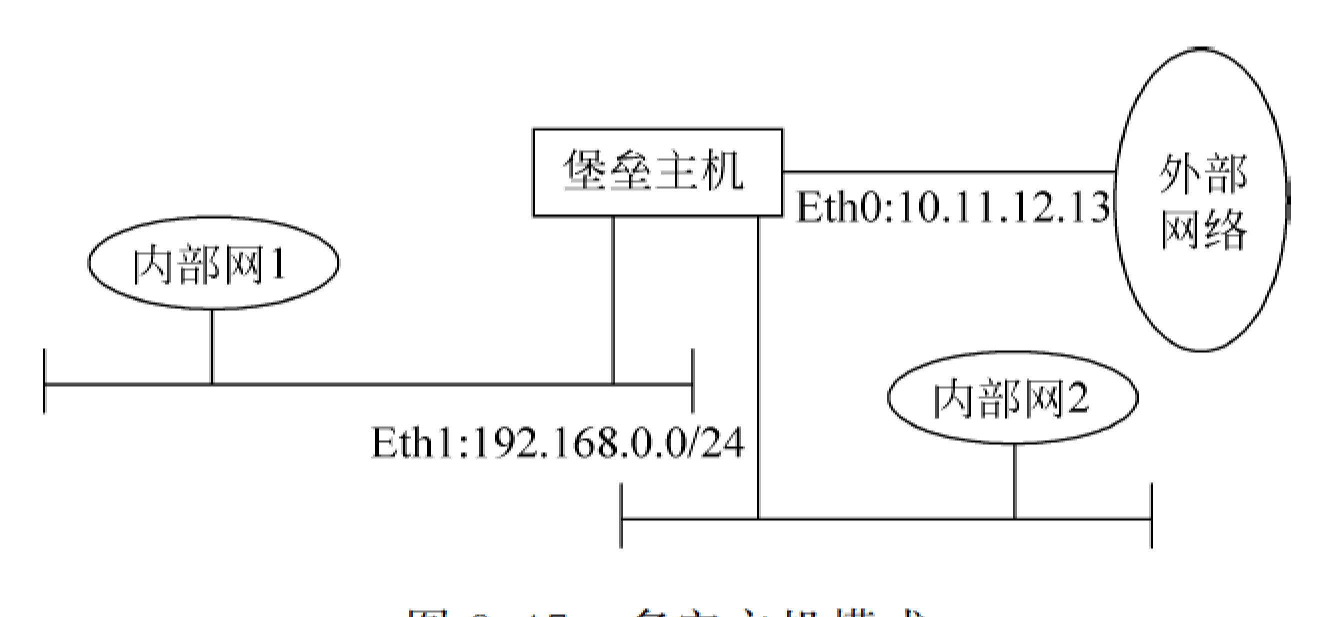


图 8.47 多穴主机模式

该防火墙的特点是多穴主机内外的网络均可以与其进行通信,但内外网络之间是不能直接进行通信的。

多穴主机防火墙的几个关键点如下。

(1) 多穴主机的 IP 转发功能必须禁止

如果多穴主机的 IP 转发功能有效，那么外部的数据包就可能绕过双穴主机防火墙进入内部网络。多穴主机可以通过代理或让用户直接注册到其上的做法提供内部和外部网络之间的通信，同时提供较高程度的安全控制能力。因此，堡垒主机上一般安装代理服务器或保存用户的登录账号信息。

(2) 多穴主机的本机安全至关重要

多穴主机是外部网络用户进入内部网络的唯一通道，也是隔开外部网络和内部网络的唯一屏障。为了保证内部网络的安全，多穴主机上应具有身份认证系统，以阻挡来自外部不可信任节点的非法登录。它的用户密码控制是一个关键，如果入侵者得到了多穴主机的控制权，内部网络就会被轻松地侵入。

(3) 多穴主机必须具备较好的性能

由于多穴主机是外部用户访问内部网络系统的中间转接点，所以它必须支持很多用户的访问，因此多穴主机的性能非常重要。

(4) 多穴主机防火墙上的用户账号数目要尽量少

因为用户的行为是不可预知的，如果多穴主机上有很多用户账号，这会给入侵检测带来很大的麻烦。用户账号的存在会给入侵者提供相对容易的入侵通道，每一个账号通常有一个可重复使用的密码，这样很容易被入侵者破解。破解密码的方法有很多，比如字典破解、强行搜索或通过网络窃听来获得。另外，如果账号太多，管理维护起来也是很费劲的，同时还会降低机器本身的稳定性和可靠性。

多穴主机防火墙有一个致命弱点，一旦入侵者侵入堡垒主机并使该主机具有路由器功能，则任何网上用户均可以随便访问有保护的内部网络。

2. 主机屏蔽模式

如图 8.48 所示，主机屏蔽防火墙的构成与包过滤防火墙、多穴主机防火墙不同，它由一台堡垒主机和一台屏蔽路由器组成。主机屏蔽防火墙实现了网络层安全(包过滤)和应用层安全(堡垒主机的代理服务)，所以入侵者在破坏内部网络的安全性之前，必须首先渗透两种不同的安全系统。

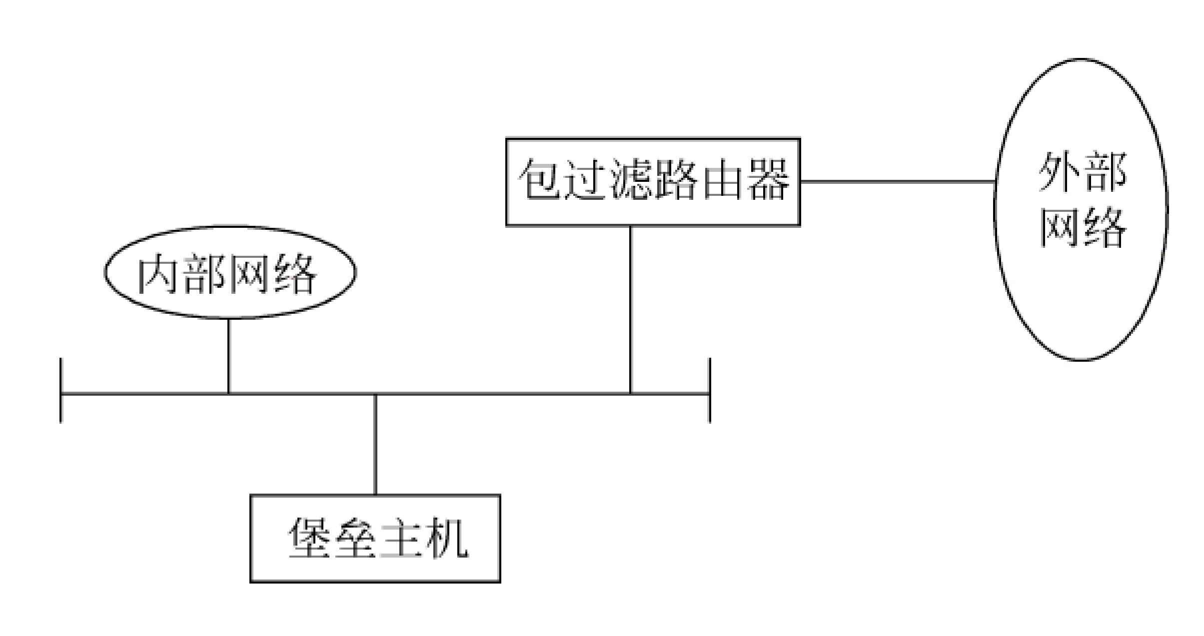


图 8.48 主机屏蔽模式

主机屏蔽防火墙模式中，提供安全保护的堡垒主机仅与内部网络相连，另有一台过滤路由器连通内部和外部网络。任何来自外部网络的连接都限制在这台堡垒主机上，内部网向外的访问可能通过该堡垒主机(例如在堡垒主机上安装代理服务器)，也可能直接经过路由器，这要取决于本地网络的安全策略。

通常在路由器上建立过滤规则，并使堡垒主机成为从 Internet 唯一可以被访问的主机，确保内部网络不受未被授权的外部用户的攻击。堡垒主机作为防火墙结构的关键点，必须具有很高的安全性。因此，堡垒主机上应该只运行必要的、经过安全改造的软件，并具有严格的审计功能。

堡垒主机是一种被强化的可以防御进攻的计算机，所谓被强化，是指它拥有较为完善的自身保护设置。这些设置基于两条原则：一是最简化原则，即在堡垒主机上设置的服务必须尽可能的少，而且对于不得不设置的服务，还要给予尽可能低的权限。因为在堡垒主机上运行的各种软件不可能不存在安全缺陷，而越是复杂的程序，就越可能含有安全缺陷。因此只有采用最简化原则，才能尽可能地减少堡垒主机的缺陷，使得堡垒主机更加安全。二是预防原则，也就是说在堡垒主机万一被攻破的情况发生时能够及时地修复它，以减少损失。为此就需要加强对堡垒主机的安全监测。

根据实际部署的需要，堡垒主机可以是单穴主机(如图 8.49 所示)，也可以是多穴主机(如图 8.50 所示)。

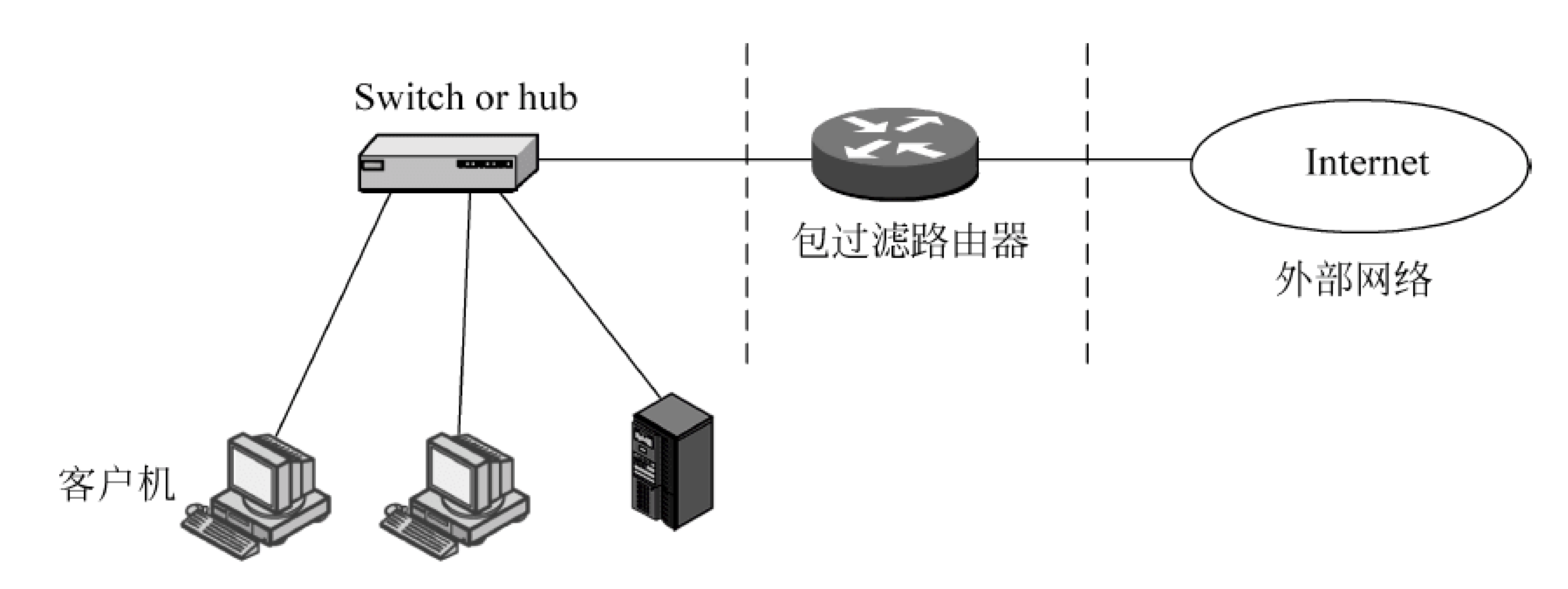


图 8.49　主机屏蔽实现方法 1

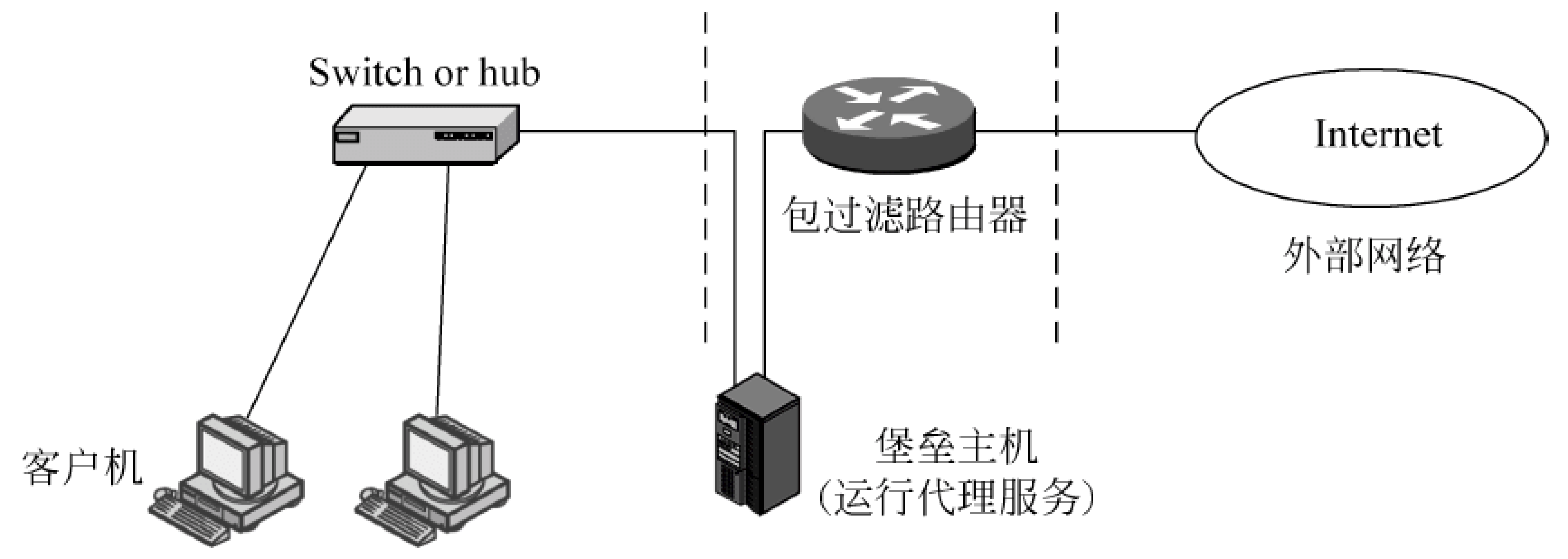


图 8.50　主机屏蔽实现方法 2

3. 屏蔽子网模式

如图 8.51 所示，子网屏蔽防火墙是指在内部网络和外部网络之间增加一个子网作为防火墙，用以保障内部网络的安全。该屏蔽子网又称为非军事区(de-militarized zone，DMZ)，或称周界网络(perimeter network)。

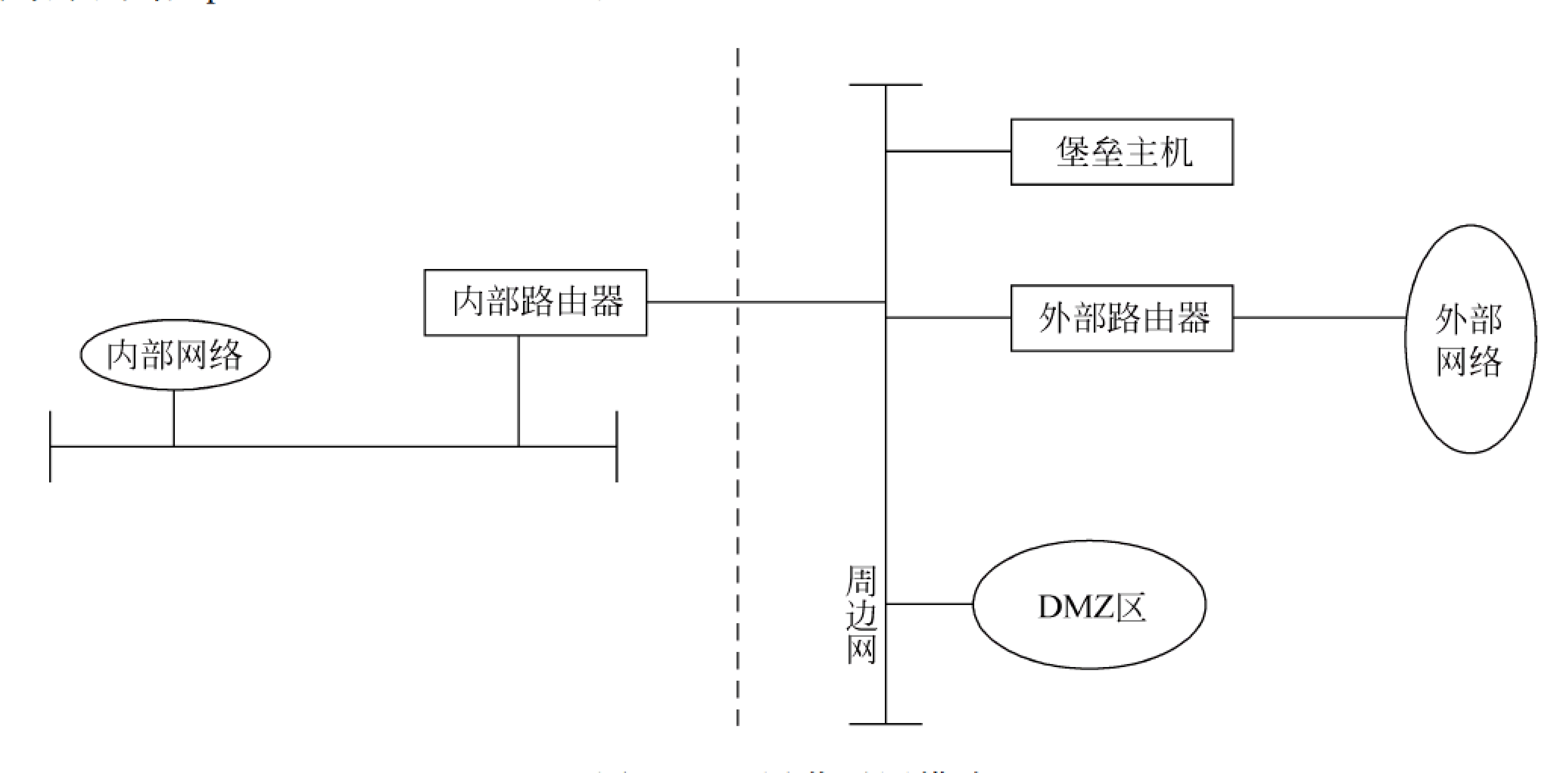


图 8.51 屏蔽子网模式

在最简单的屏蔽子网模式中，使用两台过滤路由器：一台位于过滤子网与内部网之间，另一台位于过滤子网与外部网之间。这样，整个防火墙就不会因一点被攻破而瘫痪。

屏蔽子网模式在 Intranet 和 Internet 之间建立一个被隔离的子网，用两个包过滤路由器将这一子网分别与 Intranet 和 Internet 分开。两个包过滤路由器放在子网的两端，在子网内构成一个"缓冲地带"(如图 8.51 所示)，两个路由器一个控制 Intranet 数据流，另一个控制 Internet 数据流，Intranet 和 Internet 均可访问屏蔽子网，但禁止它们穿过屏蔽子网通信。可根据需要在屏蔽子网中安装堡垒主机，为内部网络和外部网络的互相访问提供代理服务，但是来自内部和外部网络的访问都必须通过两个包过滤路由器的检查。对于向 Internet 公开的服务器，如 WWW、FTP 和 SMTP 等 Internet 服务器也可安装在屏蔽子网内，这样无论是外部用户，还是内部用户都可以访问。

根据 DMZ 的组成可以知道，子网屏蔽防火墙支持网络层和应用层安全功能，这和主机屏蔽防火墙相同。就安全性来讲，一般认为子网屏蔽防火墙的安全等级要比主机屏蔽防火墙高些。因为在主机屏蔽防火墙中内部网对堡垒主机来讲是完全公开的，如果入侵者一旦破坏了堡垒主机这层保护，那么入侵就成功了。而子网屏蔽防火墙结构是在主机屏蔽防火墙结构中再增加一台过滤路由器，这台路由器的意义就在于它能够在内部网和外部网之间构筑出一个安全子网——DMZ，这就给内部网和外部网络之间增加了一层保护。如果入侵者只侵入到 DMZ 子网中的堡垒主机，那么他只能看到 DMZ 中的信息流，而看不到内部网络中的

信息流。虽然内部网络上的数据包在内部网络上是广播式的，但内部过滤路由器会阻止这些数据包流入到 DMZ 子网中。因此即使堡垒主机受到损害也不会危及到内部网络的安全。

屏蔽子网模式限制了外部用户在内部网络中和内部用户在外部网络中的漫游能力，就像复杂地形之间的开阔地带，因而称为缓冲带或非军事区，所以这种防火墙结构又称为 DMZ 方式。

可以基于子网屏蔽防火墙的基本配置进行各种变化组合，简述如下。

(1) 合并 DMZ 的外部路由器和堡垒主机

如图 8.52 所示，这种结构是用多穴主机来执行原来外部路由器的功能。对于 WAN 速度不高的情况下，多穴主机能够胜任外部路由器的工作，只是这样多穴主机就被完全暴露在 Internet 上，因此要更加小心地保护它。

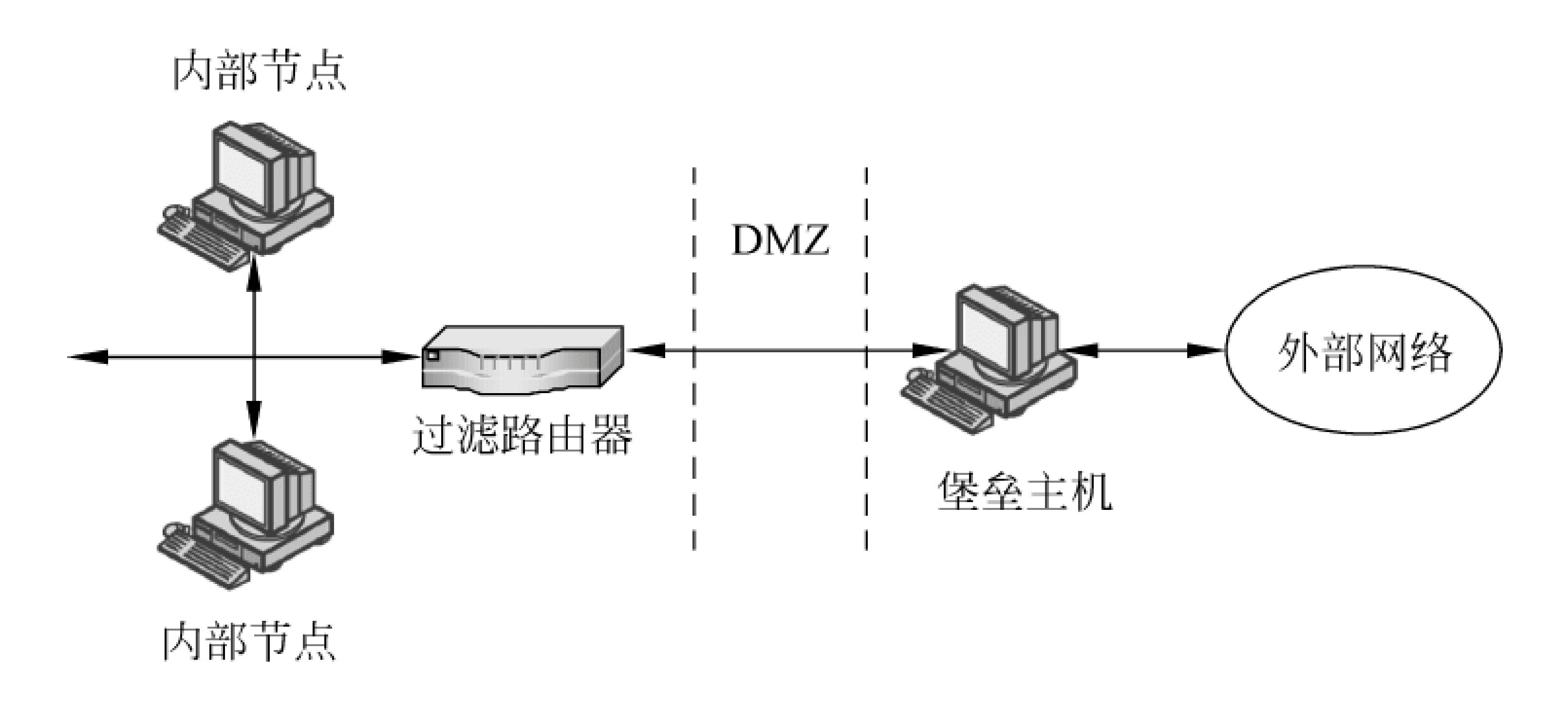


图 8.52 合并 DMZ 的外部路由器和堡垒主机

(2) 合并 DMZ 的内部路由器和外部路由器

如图 8.53 所示，这种结构同主机屏蔽防火墙一样，路由器容易受到损害。

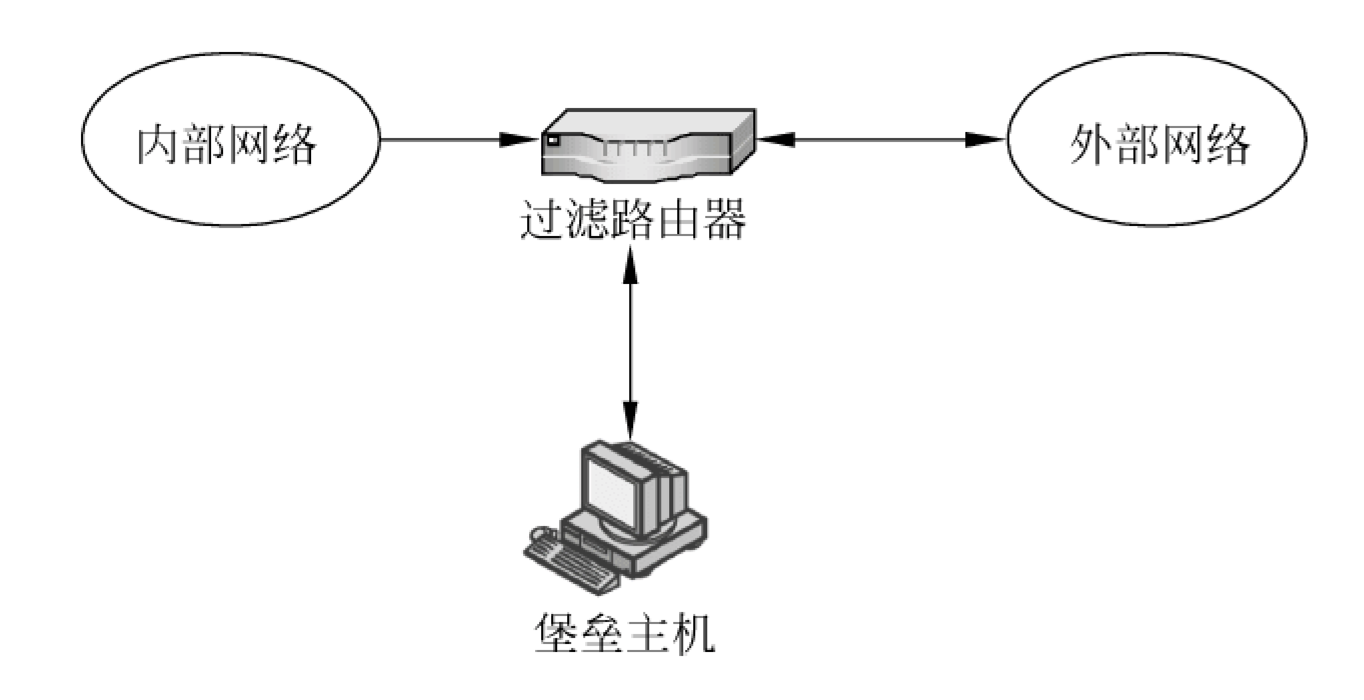


图 8.53 合并 DMZ 的内部路由器和外部路由器

(3) 多堡垒主机结构

用户在 DMZ 中使用多台堡垒主机，以提供不同的服务。

(4) 多外部路由器结构

连接多个外部路由器，使得内部网络在一个外部路由器受到损害的情况下不会受到什

么特别威胁。

(5) 其他结构

例如,可以设置多个DMZ,使每个DMZ都连接不同的外部网络,但都连接同一个内部网络。这种结构风险小,但维护相对困难。

8.4 入侵检测系统

入侵检测(intrusion detection)是指通过从计算机网络或计算机系统中若干关键点收集信息并对其进行分析,从中发现网络或系统中是否有违反安全策略的行为和遭到攻击的迹象,同时做出响应的安全技术。

入侵检测作为动态安全技术的核心技术之一,是防火墙的合理补充,也是安全防御体系的一个重要组成部分。通过入侵检测系统的部署可以扩展系统管理员的安全管理能力(包括安全审计、监视、进攻识别和响应),帮助系统监测和防范网络攻击,提高信息安全基础结构的完整性。

入侵检测技术的诞生是网络安全需求发展的必然。关于入侵检测的发展历史最早可追溯到1980年,当时James P. Anderson在一份技术报告中提出审计记录可用于检测计算机误用行为的思想,这是入侵检测的开创性的先河。另一位对入侵检测同样起着开创作用的人是Dorothy E. Denning,他在1987年提出了实时入侵检测系统模型,此模型成为后来的入侵检测研究和系统原型的基础。

入侵检测发展史上又一个具有重要意义的里程碑是NSM(network security monitor)的出现,它是由L. Todd Heberlien在1990年提出的。NSM与此前的入侵检测系统相比,其最大的不同在于它并不检查主机系统的审计记录,而是通过监视网络的信息流量来跟踪可疑的入侵行为。

8.4.1 入侵检测概述

从计算机安全的目标来看,入侵的定义是:企图破坏资源的完整性、保密性、可用性的任何行为,也指违背系统安全策略的任何事件。入侵行为不仅是指来自外部的攻击,同时内部用户的未授权行为也是一个重要的方面,内部人员滥用特权的攻击会对系统造成重大安全隐患。从入侵策略的角度来看,入侵可分为企图进入、冒充其他合法用户、成功闯入、合法用户的泄露、拒绝服务及恶意使用等几个方面。

如图8.54所示,入侵检测的一般过程是:信息收集、信息(数据)预处理、数据的检测分析、根据安全策略做出响应。

其中,信息源是指包含有最原始的入侵行为信息的数据,主要是网络、系统的审计数据

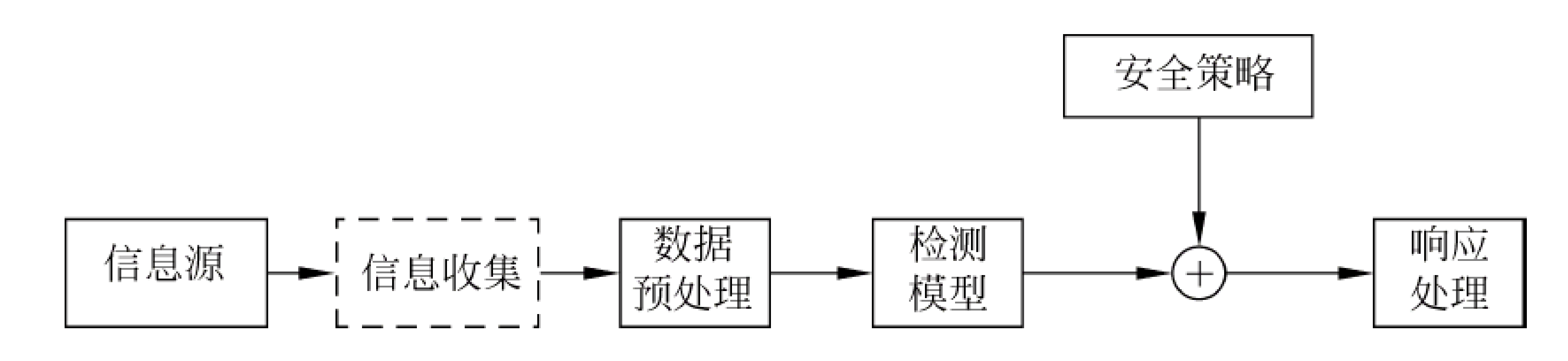


图 8.54 入侵检测的一般过程

或原始的网络数据包。数据预处理是指对收集到的数据进行预处理,将其转化为检测模型所接受的数据格式,包括对冗余信息的去除,即数据简约。这是入侵检测研究领域的关键,也是难点之一。检测模型是指根据各种检测算法建立起来的检测分析模型,它的输入一般是经过数据预处理后的数据,输出为对数据属性的判断结果,数据属性一般是针对数据中包含的入侵信息的断言。

检测结果即检测模型输出的结果,由于单一的检测模型的检测率不理想,往往需要利用多个检测模型进行并行分析处理,然后对这些检测结果进行数据融合处理,以达到满意的效果。安全策略是指根据安全需求设置的策略。响应处理主要是指综合安全策略和检测结果所做出的响应过程,包括产生检测报告、通知管理员、断开网络连接或更改防火墙的配置等积极的防御措施。

为解决入侵检测系统之间的互操作性,国际上的一些研究组织开展了入侵检测模型等相关技术的标准化工作,目前对 IDS 进行标准化工作的有两个组织:IETF 的 Intrusion Detection Working Group(IDWG)和 Common Intrusion Detection Framework(CIDF)。

CIDF 早期由美国国防部高级研究计划局赞助研究,现在由 CIDF 工作组负责,是一个开放组织。

入侵检测系统是实现入侵检测功能的一系列的软件、硬件的组合。它是入侵检测的具体实现。作为一种安全管理工具,它从不同的系统资源收集信息,分析反映误用或异常行为模式的信息,对检测的行为做出自动的反应,并报告检测过程的结果。

一般而言,入侵监测系统的主要功能应包括如下方面。

① 监视、分析用户及系统的活动。

② 检查系统配置及存在的漏洞。

③ 评估系统关键资源和数据的完整性。

④ 识别已知的攻击行为。

⑤ 统计分析异常行为。

⑥ 管理系统日志,并识别违反用户安全策略的行为。

为了达到上述目标,入侵检测系统至少应包括以下几个功能部件。

① 提供事件记录的信息源。

② 发现入侵迹象的分析引擎。

③ 基于分析引擎的结果产生反应的响应部件。

入侵检测系统就其最基本的形式来讲,可以说是一个分类器,它是根据系统的安全策略来对收集到的事件或状态信息进行分类处理,从而判断出入侵和非入侵的行为。

一般来说,入侵检测系统在功能结构上是基本一致的,均由数据采集、数据分析及响应部件等几个功能模块组成,只是具体的入侵检测系统在采集数据、采集数据的类型及分析数据的方法等方面有所不同而已。但是由于入侵技术手段的不断变化,使得入侵检测系统必须能够维护一些与检测系统的分析技术相关的信息,以使检测系统能够确保检测出对系统具有威胁的恶意事件。通常这类信息有如下几种。

① 系统、用户及进程行为的正常或异常的特征轮廓。

② 标识可疑事件的字符串,包括关于已知攻击的特征签名。

③ 激活针对各种系统异常情况及攻击行为采取响应所必需的信息。

作为新型的安全防御体系的一个重要组成部分,它的作用发挥得充分与否将在很大程度上影响整个安全策略的成败。其主要功能如下。

① 用户和系统行为的监测和分析。

② 系统配置和漏洞的审计检查。

③ 重要的系统和数据文件的完整性评估。

④ 已知的攻击行为模式的识别。

⑤ 异常行为模式的统计分析。

⑥ 操作系统的审计跟踪管理及违反安全策略的用户行为的识别。

显然,入侵检测系统完善了以前的静态安全防御技术的诸多不足,是对防火墙的合理补充,为计算机网络、系统的安全防护提供了新的解决方案。

同样,入侵检测系统作为网络安全发展史上一个具有划时代意义的研究成果,要想真正成为一种成功的产品,至少要满足以下的功能要求:实时性、可扩展性、适应性、安全性和可用性、有效性等。

8.4.2 入侵检测系统的分类

有多种方法可以对入侵检测系统进行分类,这里主要介绍两种:一种是根据入侵检测系统的输入数据来源进行的分类;另一种是根据入侵检测系统所采用的技术进行的分类。

1. 按数据来源分类

入侵检测系统的第一步是数据采集。根据入侵检测系统输入数据的来源可以将 IDS 分为基于主机的入侵检测系统和基于网络的入侵检测系统。

(1) 基于主机的入侵检测系统

基于主机的入侵检测系统(host based IDS,HIDS)通常以系统日志、应用程序日志等审计记录文件作为数据源。一般,HIDS 通过比较审计记录文件的记录与攻击签名(attack

signature，指用一种特定的方式来表示已知的攻击模式）来发现它们是否匹配。如果匹配，检测系统就向系统管理员发出入侵告警。基于主机的 IDS 可以精确地判断入侵事件，并可对入侵事件作出及时反应。它还可针对不同操作系统的特点判断出应用层的入侵事件。

利用主机数据源进行入侵检测开始并兴盛于 20 世纪 80 年代，也就是入侵检测系统发展的开始阶段。这主要是由于那时的网络及网络互连还不像现在这样广泛和复杂，网络操作系统也主要是 UNIX 系统。因此，入侵的主要手段是利用密码配置文件、远程访问配置文件和网络应用中的无密码或弱密码账户侵入网络或主机系统，利用 SUID 和 GUID 程序获取普通用户和超级用户的访问权限，或是内部人员物理侵入主机系统进行违反安全策略的活动。这些入侵手段通常会在主机的审计日志文件中留下记录，或者可以通过对主机进行基线检测来发现，而利用网络分组流则无法或难以检测到这些入侵。

目前，应用日志文件正在逐渐引起人们的关注，如 Web 服务器日志文件和数据库日志文件。早期的入侵检测系统大多都是基于主机的，它具有如下优势。

① 能够确定攻击是否成功。由于基于主机的 IDS 使用包含有确实已发生事件信息的日志文件作为数据源，因而比基于网络的 IDS 更能准确地判断出攻击是否成功。

② 适合于加密和交换环境。由于基于网络的 IDS 是以网络数据包作为数据源，因而对于加密环境难以实施入侵监测。HIDS 则不同，因为所有的加密数据在到达主机之前必须被解密，这样才能被操作系统所解析，因此它不受数据加密的限制。同时，对于交换网络来讲，基于网络的 IDS 在获取网络流量方面会面临很大的挑战，但基于主机的 IDS 没有这方面的限制。

③ 不需要额外的硬件。基于主机的 IDS 是驻留在现有网络基础设施之上的，包括文件服务器、Web 服务器和其他的共享资源等，它不需要增加新的硬件，因此入侵监测的实施成本较低。

④ 可监视特定的系统行为。基于主机的 IDS 可以监视用户和文件的访问，包括文件访问、文件权限的改变、试图建立新的可执行文件和试图访问特权服务等。例如，基于主机的 IDS 可以监视所有的用户登录及注销情况，以及每个用户连接到网络以后的行为。而基于网络的 IDS 很难做到这一点。由于操作系统记录了任何有关用户账号的添加、删除和更改等系统操作行为，基于主机的 IDS 可以监视通常只有管理员才能实施的行为，包括对系统进行的不适当的更改、影响系统日志记录的策略的变化等、对关键系统文件和可执行文件的更改等。例如，试图对关键的系统文件进行覆盖，或试图安装特洛伊木马或后门程序，这些操作都可被检测出并被终止。

基于主机的 IDS 也存在一些不足，例如，占用主机的系统资源，增加系统负荷，而且针对不同的操作系统必须开发相应的应用程序等。

（2）基于网络的入侵检测系统

基于网络的入侵检测系统（network based IDS，NIDS）以原始的网络数据包作为数据源。它利用网络适配器实时监视并分析通过网络进行传输的所有通信业务。NIDS 在进行

攻击识别时常用的技术如下。

① 模式、表达式或字节码的匹配。

② 频率或阈值的比较。

③ 事件相关性处理。

④ 异常统计检测。

作为入侵检测技术发展史上的一个里程碑，基于网络的 IDS 是网络迅速发展、攻击手段日趋复杂的新的条件下的产物。和 HIDS 比较，NDIS 具有如下特点和优势。

① 攻击者转移证据困难。NIDS 使用正在发生的网络通信进行实时攻击检测，因此攻击者无法转移证据，被 NIDS 捕获到的数据不仅包括攻击方法，而且包括对识别和指控入侵者有用的信息。由于很多黑客对系统的审计日志比较了解，因而他们知道怎样更改这些文件以藏匿他们的入侵痕迹，从这方面看，NIDS 不需要像 HIDS 那样依赖审计数据的完整性。

② 实时检测和应答。一旦发生恶意的访问或攻击，基于网络的 IDS 可以随时发现并迅速做出反应。这种实时性使得系统可以根据预先的设置迅速采取相应的行动，从而抑制入侵行为对系统的破坏力。而 HIDS 只有在可疑的日志文件产生后才能判断攻击行为，因此实时检测能力低于 NIDS。

③ 能够检测到未成功的攻击企图。有些攻击行为是指在针对防火墙后面的资源的攻击（防火墙本身可能会拒绝这些攻击企图），利用放置在防火墙外的基于网络的 IDS 就可以检测到这种企图，而基于主机的 IDS 并不能发现未能到达受防火墙保护的主机的攻击企图。通常，这些信息对于评估和改进系统的安全策略是十分重要的。

④ 与操作系统无关。NIDS 不依赖主机的操作系统，而 HIDS 需要依赖特定的操作系统才能发挥作用。

⑤ 实施方式灵活。NIDS 可以部署在网络中的一个或多个关键访问点来检测入侵行为，不需要安装在每个主机上，从而减少管理的复杂性。同时，NIDS 也可以被分布式部署和实施，从而提供更高的检测效率和更强的检测能力。

HIDS 也存在一些不足，例如，它只能监视本网段的网络活动，并且精确度较差；在交换网络环境中难于配置；防止网络欺骗的能力比较差；对于加密环境无能为力。

基于主机和基于网络的检测系统各有其自身的优点和缺陷，有些功能是不能互相替代的。现在，综合利用两种类型的数据源以获得互补特性的系统被称为混合式入侵检测系统。此外，许多其他类型的数据源，如防火墙系统、识别和认证系统、访问控制系统及其他安全设备或子系统产生的活动日志，也是未来入侵检测系统可能的数据来源。但是，由于目前尚无业界一致认可的系统和标准，因此还没有得到广泛的使用。

此外，分布式入侵检测系统也可以看作是对基于主机和基于网络的入侵检测系统的结合，它由多个部件组成，能够同时分析来自主机系统的审计数据及来自网络的数据通信信息。分布式 IDS 将是今后入侵检测系统的研究重点，它是一种相对完善的体系结构，为日

趋复杂的网络环境下安全策略的实现提供了较好的解决方案。

2. 按分析技术分类

从入侵检测的典型实现过程可以看出，数据分析是入侵检测系统的核心，它是关系到能否检测出入侵行为的关键。检测率是人们关注的焦点，不同的分析技术所体现的分析机制不同，因而其对数据分析得到的结果也不相同，而且不同的分析技术对不同的数据环境的适用性也不一样。根据入侵检测系统所采用的分析技术，可以将入侵检测系统分为采用异常检测的入侵检测系统和采用误用检测的入侵检测系统。

(1) 异常检测(anomaly detection)

异常检测也被称为基于行为的检测。其基本前提是：假定所有的入侵行为都是异常的，即入侵行为是异常行为的子集。其原理是：首先建立系统或用户的“正常”行为特征轮廓，通过比较当前系统或用户的行为是否偏离正常的行为特征轮廓来判断是否发生入侵行为，而不是依赖于具体入侵行为是否出现来进行检测的。从这个意义上来讲，异常检测是一种间接的方法。图 8.55 是典型的异常检测系统示意图。

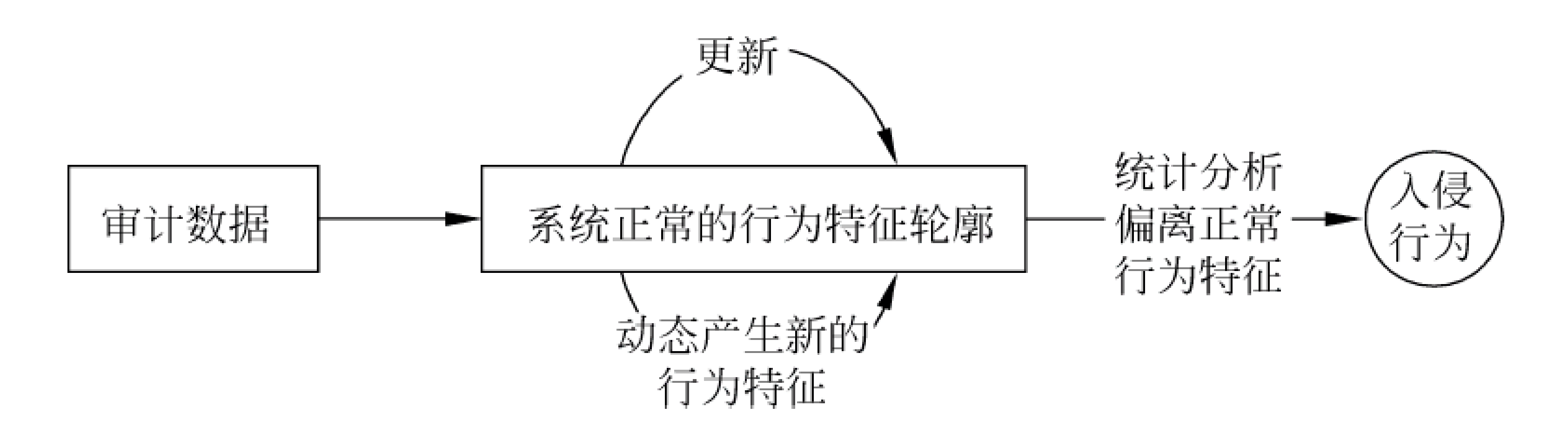


图 8.55 典型的异常检测系统示意图

从异常检测的实现机理来看，异常检测所面临的关键问题如下。

① 特征量的选择。异常检测首先是要建立系统或用户的“正常”行为特征轮廓，这就要求在建立正常模型时，选取的特征量既要能准确地体现系统或用户的行为特征，又能使模型最优化，即以最少的特征量涵盖系统或用户的行为特征。作为异常检测最关键的第一步，它将直接影响检测性能的优劣。

② 阈值的选定。因为在实际的网络环境下，入侵行为和异常行为往往不是一对一的等价关系(这样的情况是经常会有的：某一行为是异常行为，而它不一定是入侵行为；同样存在某一行为是入侵行为，而它却不一定是异常行为的情况)，所以会导致检测结果的虚警(false positives)和漏警(false negatives)的产生。由于异常检测是先建立正常的特征轮廓并以此作为比较的基准，这个基准，即阈值的选定是非常关键的。

阈值选得过大，那漏报率就会很高，这对被保护的系统的危害会很大；相反，阈值选得过小，则误报率就会提高，这会对入侵检测系统的正常工作带来很多的不便。总之，恰当地选取比较基准的阈值是异常检测的关键，是直接衡量这一检测方法准确率高低的至关重要的因素。

③ 比较频率的选取。由于异常检测是通过比较当前的行为和已建立的正常行为特征轮廓来判断入侵的发生与否的，因而比较的频率，即经过多长时间进行比较的问题也是一个重要因素。经过的时间过长，检测结果的漏警率会很高，因为攻击者往往能通过逐渐改变攻击的模式使之训练成系统能接受的行为特征，从而使攻击无法被检测出来；如果经过的时间过短，就存在虚警率提高的问题，因为有的正常的进程在短时间内的资源消耗会很大，这样检测系统就会误认为有入侵行为的发生。另外，正常的行为特征轮廓存在更新的问题，这也是在选取比较的频率时必须考虑的因素。

从异常检测的原理可以看出，该方法的技术难点在于："正常"行为特征轮廓的确定；特征量的选取；特征轮廓的更新。由于这几个因素的制约，异常检测的虚警率会很高。但对于未知的入侵行为的检测非常有效，同时它也是检测冒充合法用户的入侵行为的有效方法。此外，由于需要实时地建立和更新系统或用户的特征轮廓，因而所需的计算量很大，对系统的处理性能要求也很高。

(2) 误用检测(misuse detection)

误用检测也被称为基于知识的(knowledge-based)检测。其基本前提是：假定所有可能的入侵行为都能被识别和表示。原理是：首先对已知的攻击方法进行攻击签名表示(攻击签名是指用一种特定的方式来表示已知的攻击模式)，然后根据已经定义好的攻击签名，通过判断这些攻击签名是否出现来判断入侵行为的发生与否。这种方法是通过直接判断攻击签名的出现与否来判断入侵行为的，从这一点来看，它是一种直接的方法。图 8.56 是典型的误用检测系统示意图。

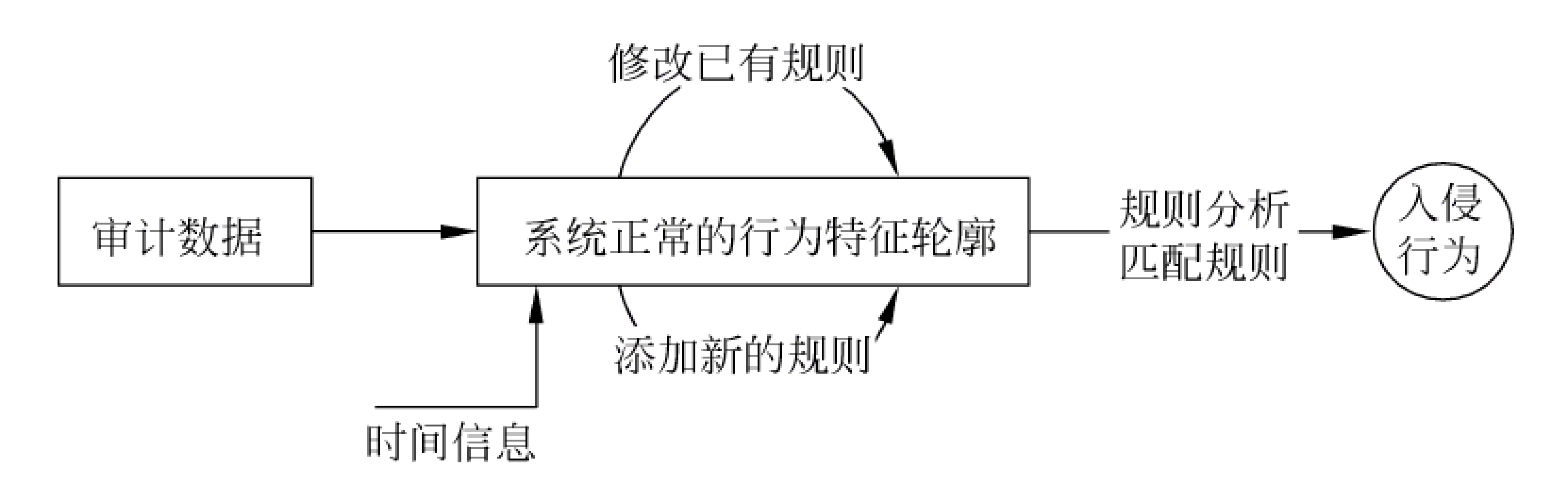


图 8.56 典型的误用检测系统示意图

同样，误用检测也存在着影响检测性能的关键问题：攻击签名的正确表示。

误用检测是根据攻击签名来判断入侵的，那么如何有效地根据对已知的攻击方法的了解，用特定的模式语言来表示这种攻击。即攻击签名的表示将是该方法的关键所在，尤其是攻击签名必须能够准确地表示入侵行为及其所有可能的变种，同时又不会把非入侵行为包含进来。

由于很大一部分的入侵行为利用的是系统的漏洞和应用程序的缺陷，因而通过分析攻击过程的特征、条件、排列及事件间的关系，就可具体描述入侵行为的迹象。这些迹象不仅对分析已经发生的入侵行为有帮助，而且对即将发生的入侵行为也有预警作用，因为只要部

分满足这些入侵迹象就意味着可能有入侵行为的发生。

误用检测是通过将收集到的信息与已知的攻击签名模式库进行比较,从而发现违背安全策略的行为的。因此,它只需收集相关的数据,这样系统的负担就明显减小了。该方法类似于病毒检测系统,其检测的准确率和效率都比较高,而且这种技术比较成熟,许多入侵检测系统都采用该方法,如 Cisco 的 NetRanger、IIS 的 Real Secure 及 Axent 公司的 IntruderAlert 等。但是其检测的完备性则依赖于攻击签名知识库的不断更新和补充。另外,误用检测是通过匹配模式库来完成检测过程的,所以在计算处理上对系统的要求不是很高。

通常,这里所能检测到的入侵行为往往是利用操作系统的缺陷、应用软件的缺陷或网络协议实现上的缺陷等来实施的。误用检测通过检测那些与已知的入侵行为模式类似的行为或间接地违背系统安全策略的行为,来识别系统中的入侵活动。使用这种技术的入侵检测系统,可以避免系统以后再次遭受同样的入侵攻击行为,而且系统安全管理员能够很容易地知道系统遭受到了哪种攻击,并采取相应的行动。但是,知识库的维护需要对系统中的每一个缺陷都要进行详细的分析,这不仅是一个耗时的工作,而且关于攻击的知识依赖于操作系统、软件的版本、硬件平台及系统中运行的应用程序等。

误用检测的主要局限性表现在如下几个方面。

① 它只能根据已知的入侵序列和系统缺陷的模式来检测系统中的可疑行为,而面对新的入侵攻击行为及那些利用系统中未知或潜在缺陷的越权行为则无能为力。也就是说,不能检测未知的入侵行为。由于其检测机理是对已知的入侵方法进行模式提取,对于未知的入侵方法由于缺乏先验知识就不能进行有效的检测,因而在新的网络环境下漏警率会比较高。

② 与系统的相关性很强,即检测系统知识库中的入侵攻击知识与系统的运行环境有关。对于不同的操作系统,由于其实现机制不同,对其攻击的方法也不尽相同,因而很难定义出统一的模式库。

③ 对于系统内部攻击者的越权行为,由于他们没有利用系统的缺陷,因而很难检测出来。

混合式入侵检测系统是对基于误用和基于异常的入侵检测方法的结合,采用这两种技术混合的方案可以做到优势互补。

除了上述对入侵检测系统的基本分类外,还有其他不同形式的分类方法,如按照入侵检测系统的响应方式来划分,可分为主动的入侵检测系统和被动的入侵检测系统。主动的入侵检测系统对检测到的入侵行为进行主动响应、处理,而被动的入侵检测系统则对检测到的入侵行为仅进行告警等。

8.4.3 入侵检测系统模型

入侵检测系统是动态安全防御策略的核心技术,比较有影响的入侵检测系统模型有 CIDF 模型;Denning 的通用入侵检测系统模型。其中,CIDF 模型是在对入侵检测系统进

行规范化的进程中提出的，也是逐渐被入侵检测领域所采纳的模型；Denning 的通用入侵检测系统模型作为入侵检测发展历程中颇具影响力的实例，给入侵检测领域的研究带来了相当重要的启示。现在，很多的入侵检测系统研究原型都是基于这两个模型的，所以有必要对其作一介绍。

1. CIDF 模型

CIDF 模型是由 CIDF（common intrusion detection framework）工作组提出的。CIDF 工作组由 Teresa Lunt 发起，是专门从事对入侵检测系统进行标准化的研究机构。它主要研究的是入侵检测系统的通用结构、入侵检测系统各组件间的通信接口问题、通用入侵描述语言（common intrusion specification language，CISL）及不同入侵检测系统间通信问题等关于入侵检测的规范化问题。

如图 8.57 所示，CIDF 提出了一个入侵检测系统的通用模型，它将入侵检测系统分为以下几个单元。

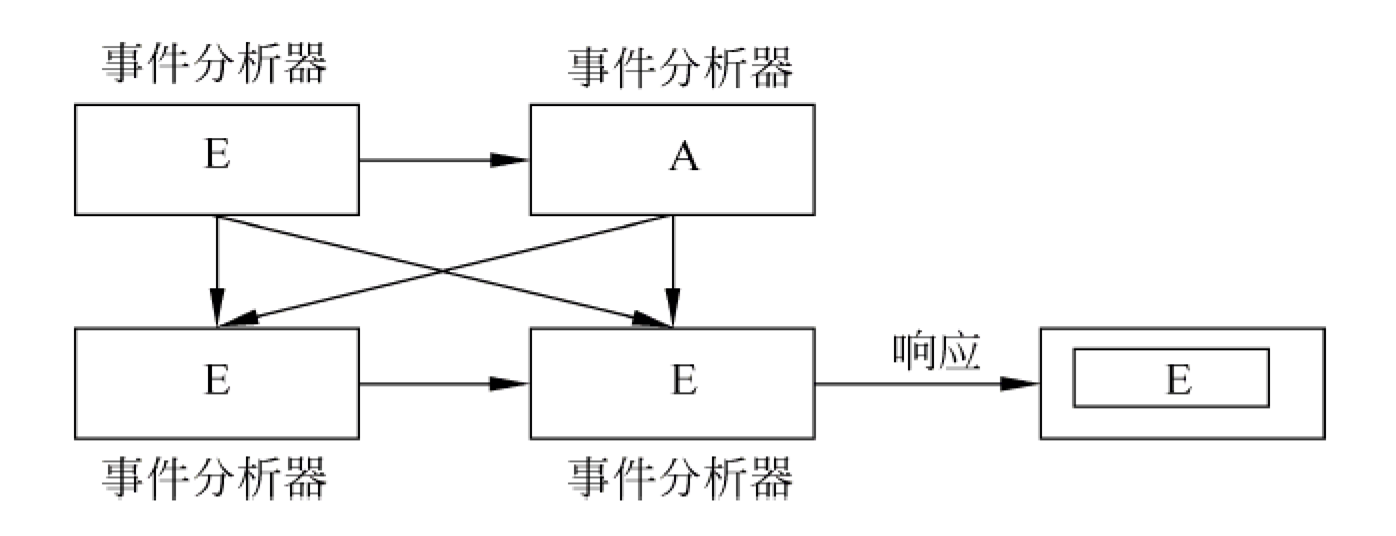


图 8.57　简化的入侵检测系统 CIDF 模型

- 事件产生器（event generators）
- 事件分析器（event analyzers）
- 响应单元（response units）
- 事件数据库（event databases）

CIDF 模型将入侵检测系统需要分析的数据统称为事件（event），它可以是网络中的数据包，也可以是从系统日志等审计记录途径得到的信息。

事件产生器即检测器，它从整个计算环境中获得事件，并向系统的其他部分提供此事件；事件分析器从分析得到数据，并产生分析结果；响应单元则是对分析结果做出反应的功能单元，它可以是做出切断连接、改变文件属性等反应，甚至发动对攻击者的反击，也可以只是简单的告警；事件数据库是存放各种中间和最终数据的地方的总称，它可以是复杂的数据库，也可以是简单的文本文件。各功能单元间的数据交换采用的是 CISL 语言。

图 8.58 是入侵检测系统的一个简化模型，它给出了入侵检测系统的一个基本框架。一般地，入侵检测系统由这些功能模块组成。在具体实现上，由于各种网络环境的差异及安全需求的不同，因而在实际的结构上就存在一定程度的差别。

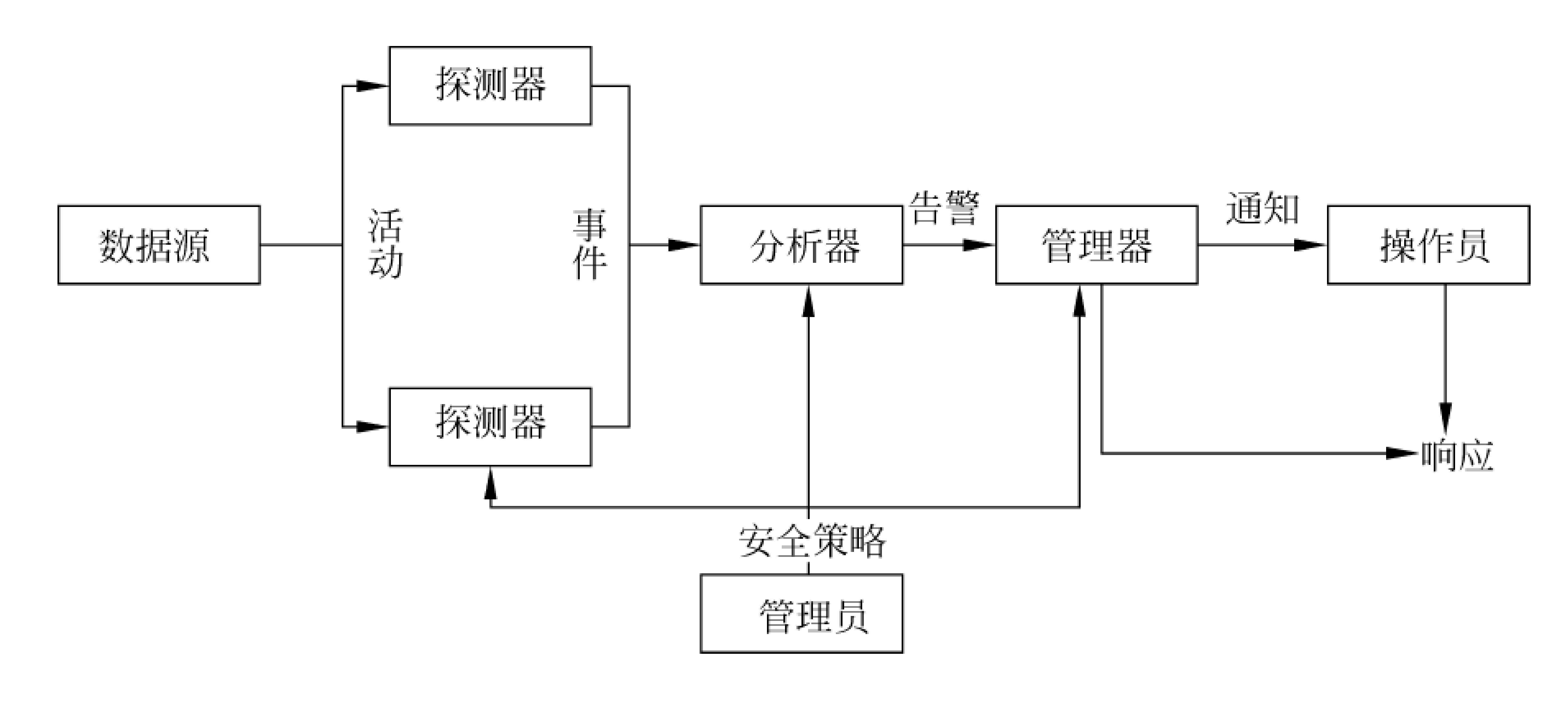


图 8.58　IETF 的入侵检测模型实例

2. Denning 通用入侵检测系统模型

如图 8.59 所示，Dorothy E. Denning 于 1987 年提出了一个通用的入侵检测模型。该模型由以下 6 个主要部分组成：主体(subjects)、客体(objects)、审计记录(audit records)、行为轮廓(profiles)、异常记录(anomaly records)及活动规则(activity rules)。

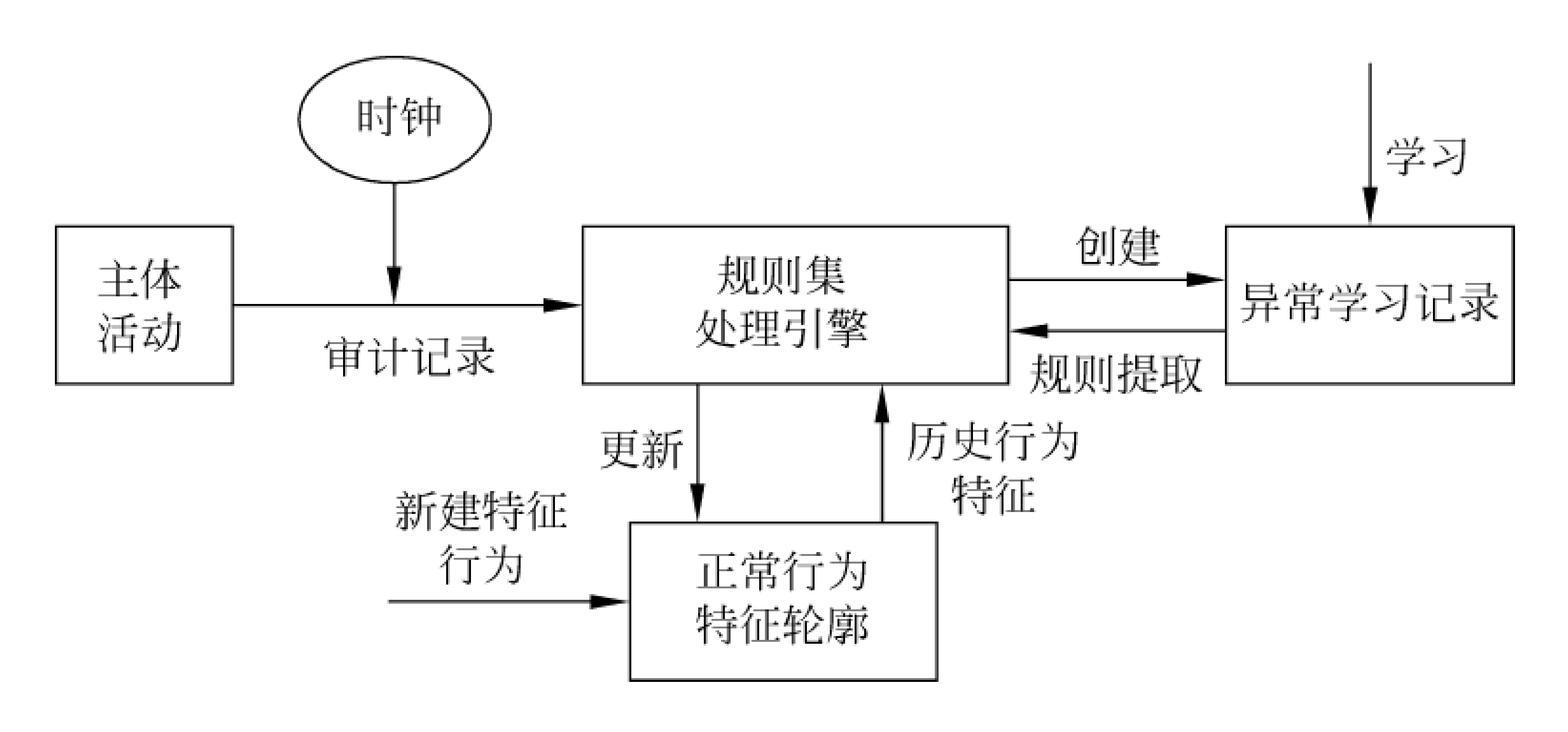


图 8.59　Denning 的通用入侵检测系统模型

该模型中，主体是指目标系统上活动的实体，通常指的是用户，也可能是代表用户行为的系统进程，或者是系统自身。主体的所有行为都是通过命令来实现的。客体是指系统资源，如文件、命令和设备等。它是主体的行为的接受者。主体和客体没有明显的界限，往往在某一环境下的主体在另一环境下则会成为客体。

审计记录是指主体对客体进行操作而在目标系统上产生的记录，如用户的登录、命令的执行及文件的访问等都会在系统中产生相应的记录。它是由＜主体，活动，客体，异常条件，资源使用状况，时间戳＞构成的六元组。其中，活动是指主体对客体的操作，如登录、退出、读和写等；异常条件是指主体活动出现异常情况时系统的状态；资源使用状况是指系统的资源消耗情况；时间戳是指活动发生的时间。

行为轮廓是描述主体对客体正常行为的模型,它包含有系统正常活动的各种相关信息。异常记录是指当系统检测到异常行为时产生的记录,由事件、时间戳和行为轮廓组成。活动规则是指系统判断是否是入侵的准则,以及当满足入侵条件时,系统所采取的相应的对策。

这个模型是个典型的异常检测的实现原型,对入侵检测的研究起着相当重要的推动作用。

SRI 的 NIDES 的异常检测器就是基于该模型的。

8.4.4 分布式入侵检测系统

就分布式入侵检测系统而言,一个大的网络环境通常是由多个入侵检测系统组成,且各个独立系统之间能够互相进行通信,或者在网络中存在一个能够对整个系统进行监控、事件分析、模块控制的中心节点。

通过各个跨越网段的分布式检测引擎,网络安全人员能够对整个网络中发生的入侵事件进行监控。分布式入侵检测系统也可以通过集中的处理攻击记录,然后通过分析,使其能够快速、容易地去检测出一个攻击的手段,发现在多个网段中存在的潜在威胁。

分布式入侵检测系统根据不同的划分标准可以有多种分类方法。

1. 根据模块控制机制分类

(1) 集中控制

对每个入侵检测模块都由一个中央控制模块来进行管理,包括对入侵检测引擎、分析器、日志管理都受控于中央控制模块。这种结构可能有分布于不同主机上的多个数据搜集器,但只有一个中央服务器,数据搜集器将当地收集到的数据踪迹发送给中央服务器进行分析处理。

探测器和管理模块可以在搭建的专用网络中进行,也可以使用现有的网络结构,探测器可以工作在混杂模式,也可以工作在非混杂模式。然而,无论在什么情况下,DIDS 都有一个显著的特征,即分布在网络不同位置的探测器都向中央控制器传送告警信息和日志信息。显然,这种结构在可伸缩性、鲁棒性和可配置性方面存在致命的缺陷:第一,随着网络规模的增加,检测引擎和服务器之间传送的数据量就会骤增,导致网络性能下降;第二,系统比较脆弱,一旦中央服务器出现故障,整个系统就会陷入瘫痪状态;第三,根据各个主机的不同需求,配置服务器非常复杂。

(2) 分散控制

分散控制分布式入侵检测系统将中央检测服务器的任务分配给多个 IDS,这些 IDS 不分等级,各司其职,负责监控当地主机的某些活动,相互之间协同工作,进行全局决策。所以,可伸缩性、安全性都得到了显著的提高,但维护成本却很高,因为针对每个 IDS 都要进行必要的维护,且分散的 IDS 之间进行数据交互很难保证通信的安全性。

2. 按照检测模块的功能分类

(1) 基于不同功能代理的分布式入侵检测系统

这是典型的DIDS的结构类型。NIDS作为探测器放置在网络的各个地方,并向中央管理平台汇报情况。攻击日志定时传动到管理平台并存在相应的存储设备当中,位于各个网段中的代理引擎具有不同的检测功能,可以把分布式入侵检测的各项检测功能均匀地分布到网络中去。这样位于网段中的检测代理因为执行了单一的功能,提高了代理的检测效率和检测的速度。当然,这样的分布式系统存在一定的不足,虽然这种结构把各个检测功能分散到各个检测代理当中,但是这样也增加了整个系统的检测风险,因为一旦某个代理出现了问题,那么系统的某个检测功能就丧失了,就会为整个分布式检测系统带来无法弥补的影响。

(2) 基于相同代理的分布式入侵检测系统

和基于不同的功能代理一样的结构类似,不同的是它的功能代理执行的检测功能是一样的,即每个检测代理具有综合的检测功能,因此它要在一个网段或者主机所有要检测的数据都要进行必要的检测,以发现是不是存在有威胁网络数据的外部攻击行为发生。相同代理的分布式入侵检测系统具有较好的稳定性,当一个检测代理失去作用的时候,其他的检测引擎也能够正常工作。但是存在的缺点就是每个检测代理需要检测的数据比较多,检测任务比较重,容易造成检测代理因为网络流量大而出现丢包的情况。

分布式入侵检测系统可以使用基于主机的入侵检测和基于网络的入侵检测相结合的技术来实现。进入20世纪90年代后,出现了把基于主机和基于网络的入侵检测结合起来的早期尝试,最早实现此种集成能力的原型系统是分布式入侵检测系统,它将NSM和Haystack组件集成到一起,并采用中央控制台来解决关联处理和用户接口的问题。

典型的DIDS是管理端/探测器结构,NIDS作为探测器放置在网络的各个地方,并向中央管理平台汇报情况。攻击日志定时地传送到管理平台,并保存在中央数据库中,新的攻击特征库则能发送到各个探测器上。每个探测器能根据所在网络的实际需要配置不同的规则集。告警信息能发送到管理平台的消息系统,用各种方式通知IDS管理员。现在有人根据此种数据来源于混合的方式,称此类系统为"混合型(hybrid)"系统。最著名的明确体现分布式架构的早期系统为SRI的EMERALD系统,它明确将分布式检测架构进行层次化的处理,并实现了不同层次上的分析单元,同时提供了开放的API接口,实现基本架构下的组件互换功能。之后,UC Davis设计了GrIDS(graph-based IDS)系统,这也是处理可扩展性问题的一次有益尝试。后来的Purdue大学设计并原型实现的AAFID系统体现了基于自治代理的分布式架构思想。

典型的分布式入侵检测系统采用基于主机的和基于网络的入侵检测系统相结合的综合方案,这样既可以克服基于主机的入侵检测系统和基于网络的入侵检测系统的各自不足,又可以充分发挥它们各自的优势,从而实现对被保护目标的最佳防护。图8.60是分布式入侵

检测系统的组成框图。

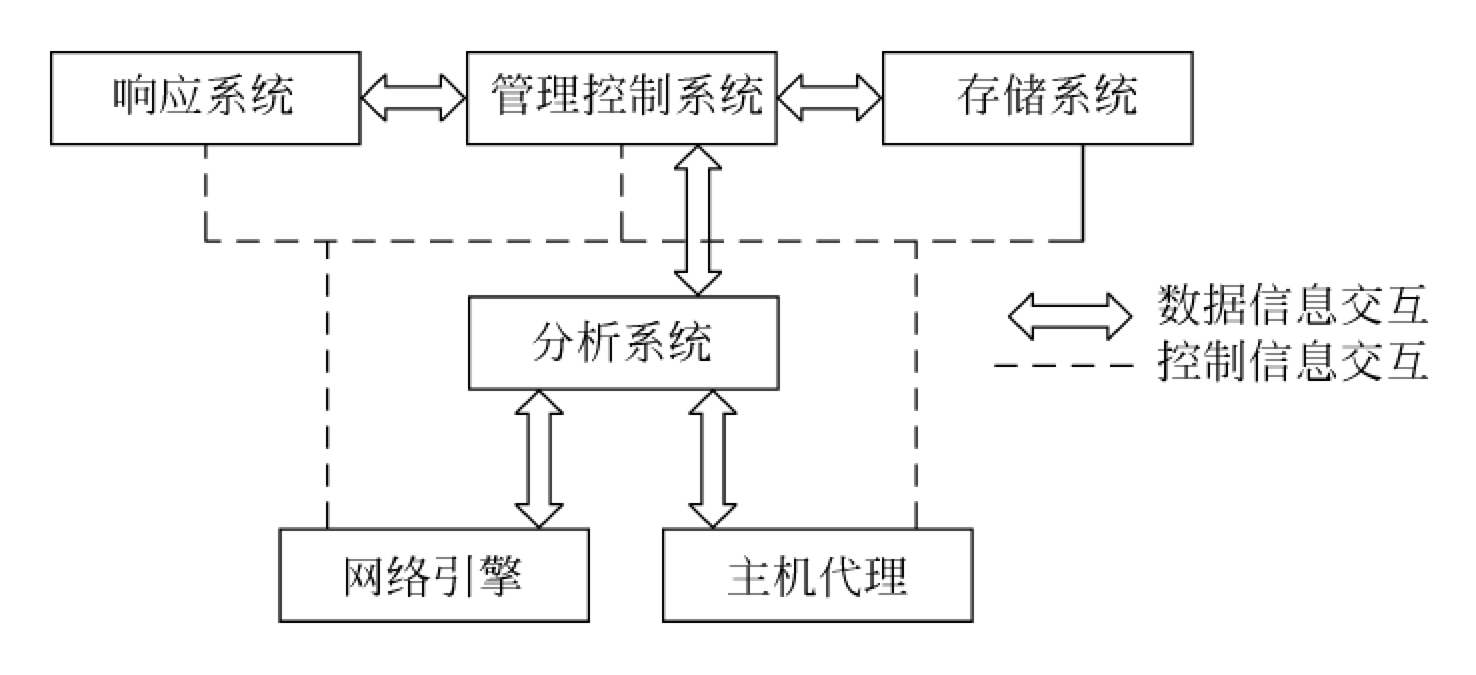


图 8.60　分布式入侵检测系统设计框图

由图 8.60 可以看出，该系统的主要功能部件有网络引擎、主机代理、分析系统、管理控制系统、存储系统和响应系统等。网络引擎主要是从网络流量中获取原始数据包，并对其进行预处理，并将预处理后的数据发送给分析系统；主机代理则是从受保护的主机系统获取审计数据，并对其进行预处理，将处理过的数据送往分析系统；分析系统对预处理后的数据进行分析，根据不同的数据特点建立相应的检测模型，即采用不同的检测算法对数据进行分析处理，并将分析结果送到管理控制系统；管理控制系统是整个系统同用户交互的窗口，它提供各种管理控制信息，并协调其他部件的工作；存储系统是用来对各种结果进行存储的地方，并提供灵活的数据维护、处理和查询服务，同时也是一个安全的日志系统；响应系统则是对确认的入侵行为采取相应措施的子系统。

从所采用的技术角度来看，分布式入侵检测系统的检测机制是误用检测和异常检测并举的方案。具体的工作模式如图 8.61 所示。

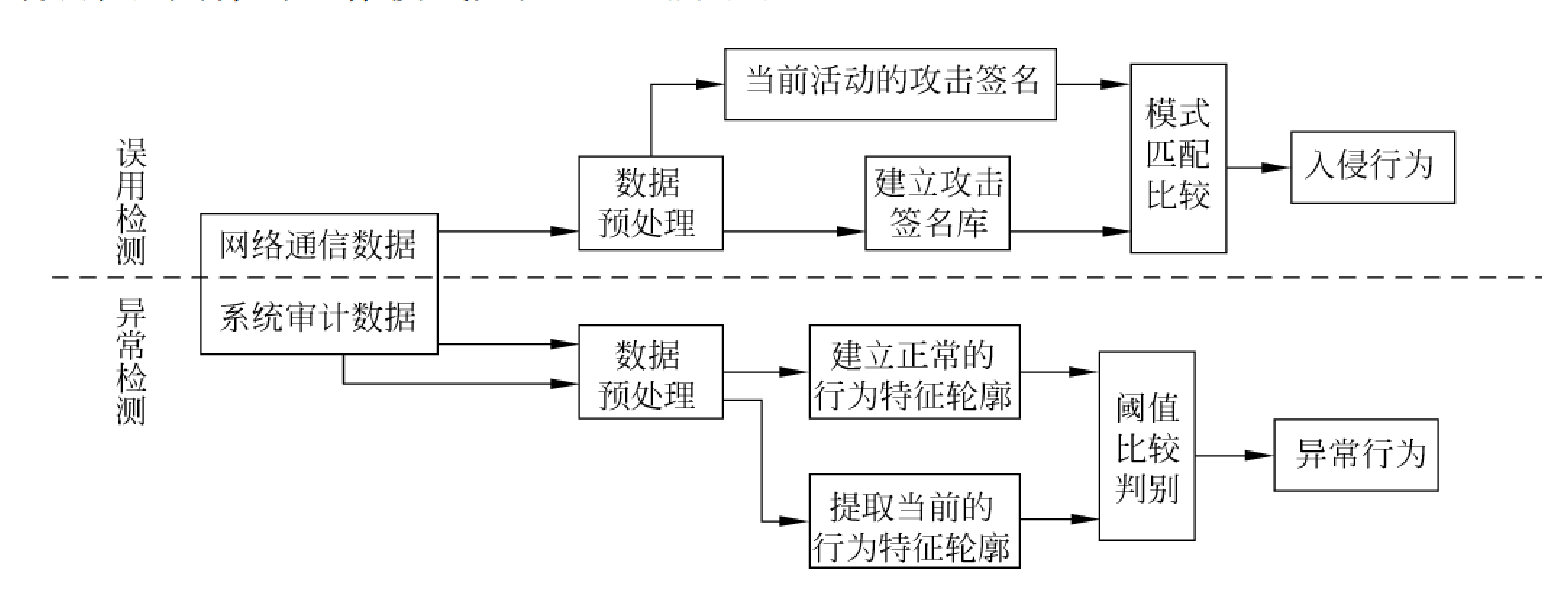


图 8.61　分布式入侵检测系统的检测机制示意图

图 8.62 给出了一个典型的分布式入侵检测系统在实际网络环境下的部署图。在本图的防火墙内外都设置了网络引擎，这样就可以充分利用基于网络的 IDS 的优点，实时地进行攻击企图的识别，并可将其阻断在防火墙之外。同时，还可监控透过防火墙的攻击行为，

为我们及时地更新防火墙的配置提供依据。对于主机代理的设置则要根据具体的安全防护策略来进行。这样,就可以最大限度地发挥不同类型的 IDS 的优势,实现对被保护网络的最大安全化。

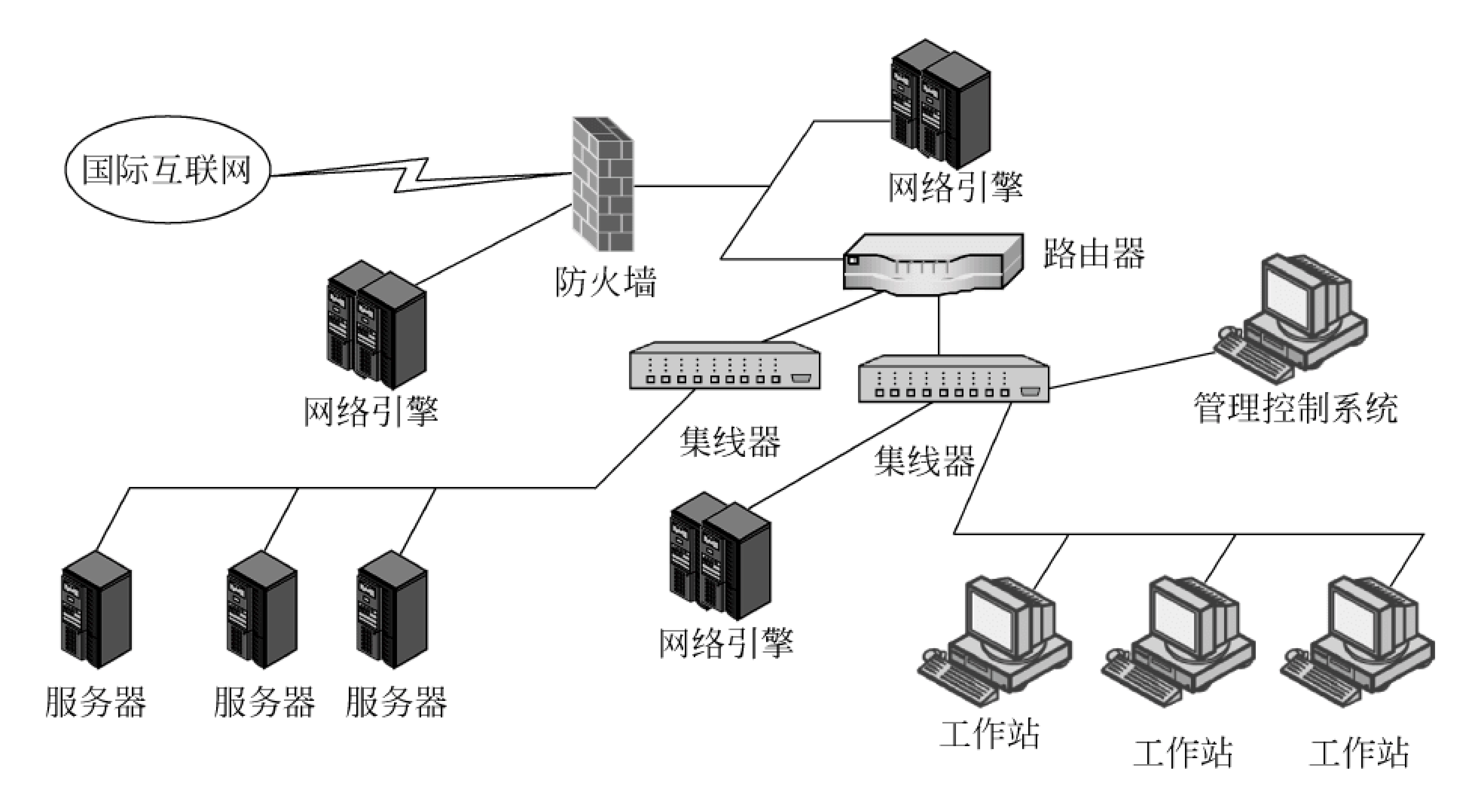


图 8.62 典型的分布式入侵检测系统部署图

8.4.5 SNORT 入侵检测系统

1. SNORT 概述

Snort 是一种基于 libpcap 的数据包嗅探器,可以作为一个轻量级的网络入侵检测系统。所谓的轻量级是指在检测时尽可能低地影响网络的正常操作。Snort 是一种基于网络的 IDS,并且 Snort 可以运行于多种操作系统平台,例如 UNIX 系列和 Windows 系列。与很多商业产品相比,它对操作系统的依赖性比较低。其次,用户可以根据自己的需要及时地在短时间内调整检测策略。

Snort 集成了多种告警机制来提供实时告警功能,包括 syslog、用户指定文件、UNIXSocket、通过 SMBClient 使用 WinPopup 对 Windows 客户端告警等。

Snort 是开放源代码的,通过适当配置,可以帮助中小网络的系统管理员有效地监视网络流量和检测入侵行为。

Snort 系统的基本功能包括数据包嗅探、数据包记录和入侵检测等。同时,通过配置,Snort 还能够完成更复杂的功能。Snort 取得数据包后先用预处理器插件处理,然后经过检测引擎中的所有规则链,如果有符合规则链的数据包,就会被检测出来。Snort 的预处理器、检测引擎和告警模块都是插件结构,插件程序按照 Snort 提供的插件函数接口完成,使用时动态加载,在使用修改核心代码的前提下让 Snort 的功能和复杂性扩展更容易。这既

保障了插件程序和 Snort 的核心代码的紧密相关性，又保障了核心代码的良好扩展性。

下面介绍 Snort 的基本功能模块。

（1）数据包嗅探器

Snort 的最基本功能就是数据包嗅探。然而，数据包嗅探只是 Snort 工作的开始。在因特网上通常指嗅探 IP 网络的流量，但是对不常用的网络协议如 IPX 和 AppleTalk 等也能嗅探。在 IP 数据包中包含了不同类型的协议，如 TCP、UDP、ICMP、IPSec 和路由协议等，因此很多数据包嗅探器还会做协议分析，把分析结构展现出来。数据包嗅探器有以下几种用法。

- 网络分析和网络故障查找。
- 网络性能和负荷量分析。
- 监听明文传输的用户名密码等敏感数据。

（2）预处理器

基于特征/规则匹配的 IDS 系统因为速度快而受到众多用户的信赖。这种检测系统的缺点是：如果攻击模式很常见，就会产生很多误报。如果模式过于特殊，又会产生漏报。造成这些缺陷的原因是特征语言的表达能力有限或 IDS 对协议的分析不够，一些 IDS 通过复杂的方法解决这一问题。Snort 则通过预处理器来实现这些功能。这些功能主要包括。

- 包重组。
- 协议解码。
- 异常检测。

（3）检测引擎模块

检测引擎是 Snort 的核心模块。当数据包从预处理器送过来后，检测引擎依据预先设置的规则检查数据包，一旦发现数据包中的内容和某条规则相匹配，就通知告警模块。

（4）告警/日志模块

经检测引擎检查后的 Snort 数据需要以某种方式输出，如果检测引擎中的某条规则被匹配，则会触发一条告警，这条告警信息会记录在日志文件中。告警信息也可记入数据库中。

2. SNORT 体系结构

SNORT 系统的模块构成及各模块之间的关系如图 8.63 所示。

- 主控模块：实现的功能包括模块的初始化、命令行解释、配置文件解释、数据包捕获库 wincap 的初始化；调用 libpcap 捕获数据包，进行解码检测入侵；对系统中的插件进行管理（包括插件的初始化和启动）。
- 解码模块：把从网络上抓取的原始数据包沿各个协议栈进行解码，填充相应的数据结构，以便由规则处理模块进行处理。
- 规则处理模块：实现对从解码模块得到的各种报文进行子规则的模式匹配，检测出

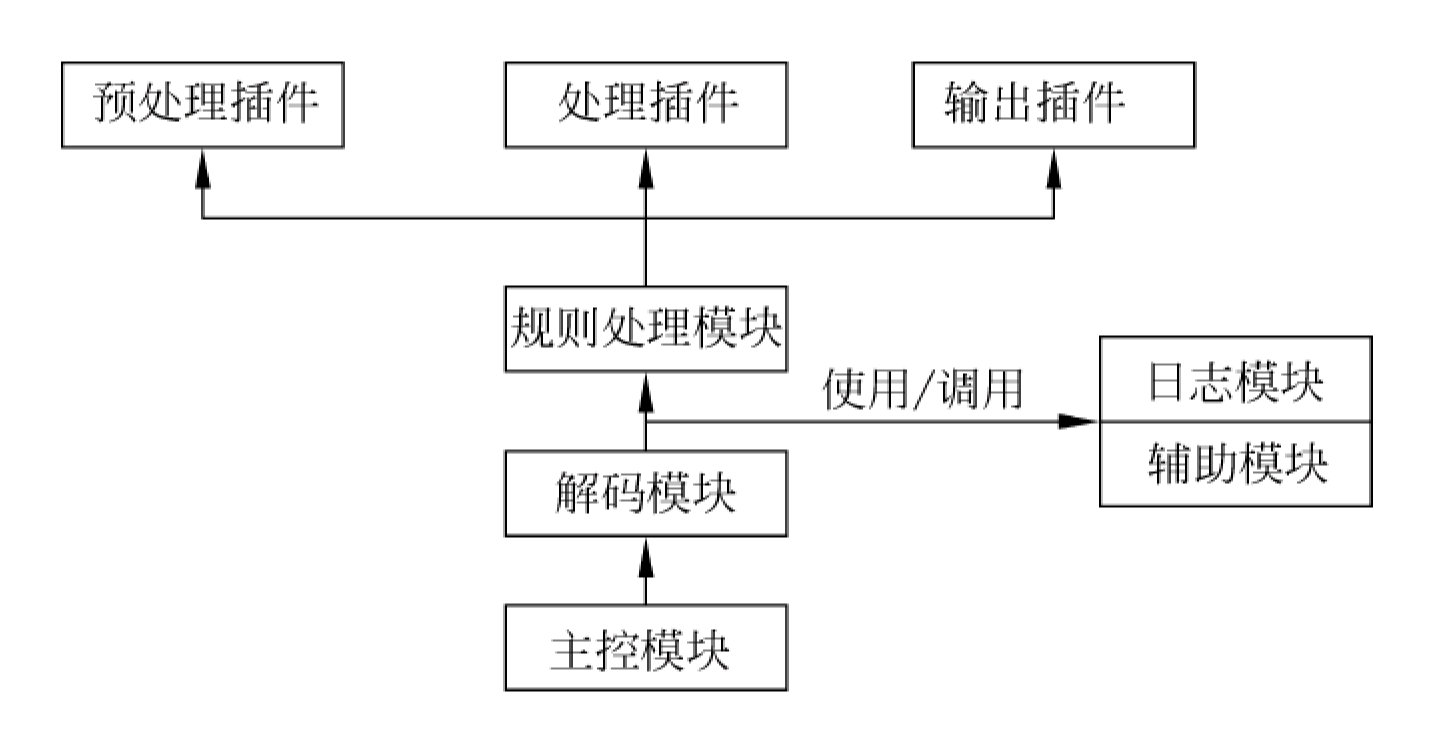


图 8.63 SNORT 系统模块构成

入侵行为。在初始化阶段，它还负责完成规则文件的解释和规则语法树的构建工作。规则处理模块在执行检测工作时使用 3 种形式的插件，分别为预处理插件模块、处理插件模块和输出插件模块。

- 预处理器插件：在模式匹配之前进行，对报文进行分片重组、流重组和异常检查等预处理操作。
- 处理插件：对数据包进行检查，包括数据包的大小、协议类型及 IP/ICMP/TCP 的选项等，辅助规则匹配完成检测功能。
- 输出插件：实现在检测到攻击后执行各种输出和反应的功能。
- 日志模块：实现各种报文日志功能，即把各种类型的报文记录到各种类型的日志中。

在系统运行过程中，还用了一些辅助模块，如树结构定义子模块定义了几种 Snort 使用到的二叉树结构和相关的处理函数；tag 处理模块完成和 tag 模块相关的功能。另外，一些子模块提供了一些公用的函数，如字符处理等。

3. 入侵检测流程

基于规则的模式匹配是 Snort 的核心检测机制。Snort 的入侵检测流程分两步：首先是规则的解析流程，包括从规则文件中读取规则和在内存中组织规则；其后是使用这些规则进行匹配的入侵检测流程。

(1) 规则解析流程

Snort 的规则解析流程很简单：首先读取规则文件，紧接着一次读取每一条规则；然后对其进行解析，用相应的规则语法表示，在内存中对规则进行组织，建立规则语法树。

所有的规则按照规则头排成主链，然后根据规则选项把规则插入到这个链中，构成一棵规则树，这样每一个选项节点就对应一条规则。规则头节点主要记录了规则头信息，包括源 IP 端口、目标 IP 端口，并有一指针指向下一个规则头节点，附以自身规则选项列表和规则头列表结构。规则选项节点存放所有的规则选项的信息和处理插件的处理函数列表，分别

指向规则头节点的指针和关联选项节点的指针。

(2) 规则匹配流程

规则匹配的流程就是对从网络上捕获的每一条数据报文和上面描述的规则树进行匹配的过程。如果发现存在一条规则匹配这个报文,就表示检测到一个攻击,然后按照规则指定的行为进行处理(如发送警告等);如果搜索完所有的规则都没有找到匹配的规则,就表示报文是正常的报文。

所有的规则被组织成规则树,然后分类存放在规则类列表中。总体的检测过程归根结底是对规则树进行匹配扫描,找到报文所对应的规则。对规则树的匹配过程则是先根据报文的 IP 地址和端口号,在规则头链表中找到相对应的规则头,找到后再接着匹配此规则头附带的规则选项链表。

4. SNORT 规则

SNORT 规则是基于文本的,它通常存在于 Snort 程序目录或者子目录中。规则文件按照不同的组进行分类,比如文件 ftp. rules 包含了 FTP 攻击内容。在启动的时候,Snort 读取所有的规则文件,并且建立一个三维的链表。Snort 使用该三维链表匹配包和实现检测。

主文件 snort. c 进行初始化时建立三维链表,启动时 Snort 系统读取 snort. conf 配置文件,在 snort. conf 文件中链接特定的规则文件,如 snort. conf 文件的部分内容如下所示。

```
include $RULE_PATH/bad-traftic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/rpc.rulcs
```

主函数将每个规则文件中的 Snort 规则解析,在内存中建立用来进行模式匹配的数据结构。

(1) 规则格式

Snort 规则可以划分为两个逻辑部分:规则头(rule header)和规则选项(rule options)。

规则头如图 8.64 所示。规则头包含了规则动作、协议、IP 源地址和目的地址,子网掩码及源端口和目标端口值等信息。而规则选项则包含警报信息及用于确定是否触发响应规则动作而需检查的数据包区域位置的相关信息。



图 8.64 Snort 规则头

规则选项并不是对每一个规则都是必需的,它们

只是用来更好地定义所要进行某种处理(如记录、警报等)的数据包类型。只有当规则中的每一个元素都为真时才能触发对应的规则动作,即规则元素之间形成的是一种逻辑“与”的关系。与此同时,在每个规则库文件中的各种规则集合之间形成的是一种更大范围上的逻辑“或”关系。

(2) Snort 规则树

规则树由链表头和规则树节点(rule tree nodes,RTN)及选项树节点(option tree nodes,OTN)组成。

① 链表头。Snort 规则树有 5 个单独的规则链,这些链作为“树”顶部的链表头。

- Activation:告警并且开启另外一个动态规则。
- Dynamic:当被上层的激活规则调用时,记录网络流量的日志。
- Alert:产生告警并记录这个数据包。
- Pass:忽略这个数据包。
- Log:记录网络的流量(不告警)。

② 规则树节点和选项树节点。对于 5 个规则链中的每一个,都有单独的被协议关闭的链表,树中的这一层被称为规则树节点。规则树节点支持以下 4 个协议。

- TCP TCP 协议:如 SMTP、HTTP 和 FTP。
- UDP UDP 协议:如 DNS lookups。
- ICMP ICMP 协议:如 ping、traceroute。
- IP IP 协议:如 IPSec、IGMP。

每一个协议链表中的是规则选项,称为选项树节点。

其中,Content 中的内容是具体采用 BM 或 AC 模式匹配算法进行查找的特征字符串。Flow 中的内容是系统根据设置,指定要链接到的检测插件。初始化时,SNORT 读入规则文件,把规则集合组织成一个二维的链表结构,这个链表结构包含规则树节点和选项树节点。

规则树节点中包含规则的通用属性,例如源 IP 地址、源端口号、目的 IP 地址、目的端口号和协议类型(TCP、ICMP、UDP)等。选项树节点中包含一些可被添加到每条规则中的各种各样的信息,例如 TCP 标志、ICMP 代码、类型、包负载的大小,影响效率的主要瓶颈,要查找的内容等。

RTN 节点从左到右组成一个链,并作为各个 OTN 链的链头,即 OTN 链是链在与之相关的 RTN 节点的下面的。按照给定的规则集对包进行检查时,首先沿着 RTN 链从左向右进行匹配直到找到一个匹配的 RTN 节点。当要检查的包与某一个 RTN 节点匹配时,沿着链在它下面的 OTN 链继续向下查找,对每个 OTN 中的选项的检查采用相应的插件函数进行,这些插件函数也同样被组织成链表的形式。当 OTN 节点中的一个选项与包匹配时,当前的插件函数调用链表中的下一个插件函数对该 OTN 节点中下一个选项进行检查。如果一个选项检查失败,则跳出该 OTN 节点,对 OTN 链表中的下一个 OTN 节点进行检查。

为了提高效率，先对不需要对包内容进行检查的选项进行检查，然后再对需要对包内容进行检查的选项进行检查，以减少不必要的匹配所需的计算量。如果需要对包的内容进行检查，则使用著名的BM算法或经过优化的AC算法，将OTN节点选项所要求检查的模式串与包的内容进行精确的模式匹配。如果包中没有包含要找的串，继续与链表中下一个OTN节点中的选项所要查找的模式串进行匹配，直到在包中找到所要查找的内容或所要查找的串全部查找一遍为止。

8.4.6 入侵检测的发展趋势

无论在规模上还是方法上说，入侵技术近年来都发生了变化。入侵的手段与技术也有了很大的发展和演化，主要反映在下列几个方面。

① 入侵或攻击的综合化与复杂化。入侵的手段有多种，以往的入侵者往往采取一种攻击手段。由于当前网络防范技术的多重化，攻击的难度增加，使得入侵者在实施入侵或攻击时往往同时采取多种入侵的手段，以保证入侵的成功概率，并可在攻击实施的初期掩盖攻击或入侵的真实目的。

② 入侵主体对象的间接化，即实施入侵与攻击的主体的隐蔽化。通过一定的技术，可掩盖攻击主体的源地址及主机位置。即使用了隐蔽技术后，对于被攻击对象来说，真正的攻击主体是无法直接确定的。

③ 入侵或攻击的规模日益扩大。对于网络的入侵与攻击，在其初期往往是针对于某公司或一个网站，其攻击的目的可能为某些网络技术爱好者的猎奇行为，也不排除商业的盗窃与破坏行为。由于战争对电子技术与网络技术的依赖性越来越大，随之产生、发展、逐步升级到电子战与信息战。对于信息战，无论其规模与技术都与一般意义上的计算机网络的入侵与攻击不可相提并论。信息战的成败与国家主干通信网络的安全是与任何主权国家领土安全一样重要的国家安全。

④ 入侵或攻击技术的分布化。以往常用的入侵与攻击行为往往由单机执行。由于防范技术的发展使得此类行为不能奏效。所谓的分布式拒绝服务在很短时间内可造成被攻击主机的瘫痪。且此类分布式攻击的单机信息模式与正常通信无差异，所以往往在攻击发动的初期不易被确认。分布式攻击是近期最常用的攻击手段。

⑤ 攻击对象的转移。入侵与攻击常以网络为侵犯的主体，但近期的攻击行为却发生了策略性的改变，由攻击网络改为攻击网络的防护系统，且有越演越烈的趋势。现已有专门针对IDS做攻击的报道。攻击者详细地分析了IDS的审计方式、特征描述、通信模式并找出IDS的弱点，然后加以攻击。

相应的，入侵检测技术主要以下面这些为发展方向。

- 分布式入侵检测：针对分布式网络攻击的检测方法，使用分布式的方法来检测分布式的攻击，其中的关键技术为检测信息的协同处理与入侵攻击的全局信息的提取。

- 智能化入侵检测：即使用智能化的方法与手段来进行入侵检测。所谓的智能化方法，现阶段常用的有神经网络、遗传算法、模糊技术和免疫原理等方法，这些方法常用于入侵特征的辨识与泛化。利用专家系统的思想来构建入侵检测系统也是常用的方法之一。特别是具有自学习能力的专家系统，实现了知识库的不断更新与扩展，使设计的入侵检测系统的防范能力不断增强，应具有更广泛的应用前景。应用智能体的概念来进行入侵检测的尝试也已有报道，较为一致的解决方案应为高效常规意义下的入侵检测系统与具有智能检测功能的检测软件或模块的结合使用。

全面的安全防御方案：即使用安全工程风险管理的思想与方法来处理网络安全问题，将网络安全作为一个整体工程来处理。从管理、网络结构、加密通道、防火墙、病毒防护和入侵检测多方位全面对所关注的网络作评估，然后提出可行的全面解决方案。

本章实验

1. Windows 系统 VPN 的配置。
2. Linux 防火墙的配置。
3. 安装调试入侵检测工具 Snort。

思考题

1. 假设公司拥有一个 PPTP VPN，公司雇员出差在外地通过 ADSL 拨入公司内部网络，这种情况下将有两个 PPP 连接被建立，仔细考虑一下这两个 PPP 连接分别是如何建立的？数据包的封装又是怎样的？私有地址通过哪个连接分配给雇员的客户端机器？

2. MPLS BGP VPN 是如何利用虚拟路由器隔离 VPN 之间的路由的？MPLS VPN 还有什么其他的方式？

3. 基于异常的入侵检测和基于误用的异常检测各有什么优点和缺点？

4. MAC、DAC 和 RBAC 的交集存在吗？Windows 系统中采用的是什么访问控制机制？

5. 一台 Linux 主机上安装了代理服务器软件，同时它也作为带有网络地址转换功能的路由器存在于内部网络和外部网络之间，在收到内部网络中访问 Internet 的请求后这两个功能分别如何处理该请求？Wingate、Sygate 及 CCproxy 分别属于哪种角色？

第 9 章

无线网络及移动IP安全

9.1 无线网络安全概述

9.1.1 无线网络及其分类

目前无线接入网的总体框架是一个由混合技术构建的、基于互连固定基站(或接入点AP)和蜂窝结构的体系结构,如图9.1所示。其中,有线结构部分由交换机、路由器和支持无线网络运行的移动管理部件组成。未来的无线通信网络将包含下列部分:地面蜂窝(terrestrial cellular/PCS),高容量链接(有线的或固定的点对点无线通信),可编程的多波段多方式无线通信和连接基础设施的或以ad hoc方式运行的高速WLAN。提供信息的设备包括服务器、台式机、笔记本式计算机、PDA、蜂窝电话和传感器等。移动节点MN(mobile

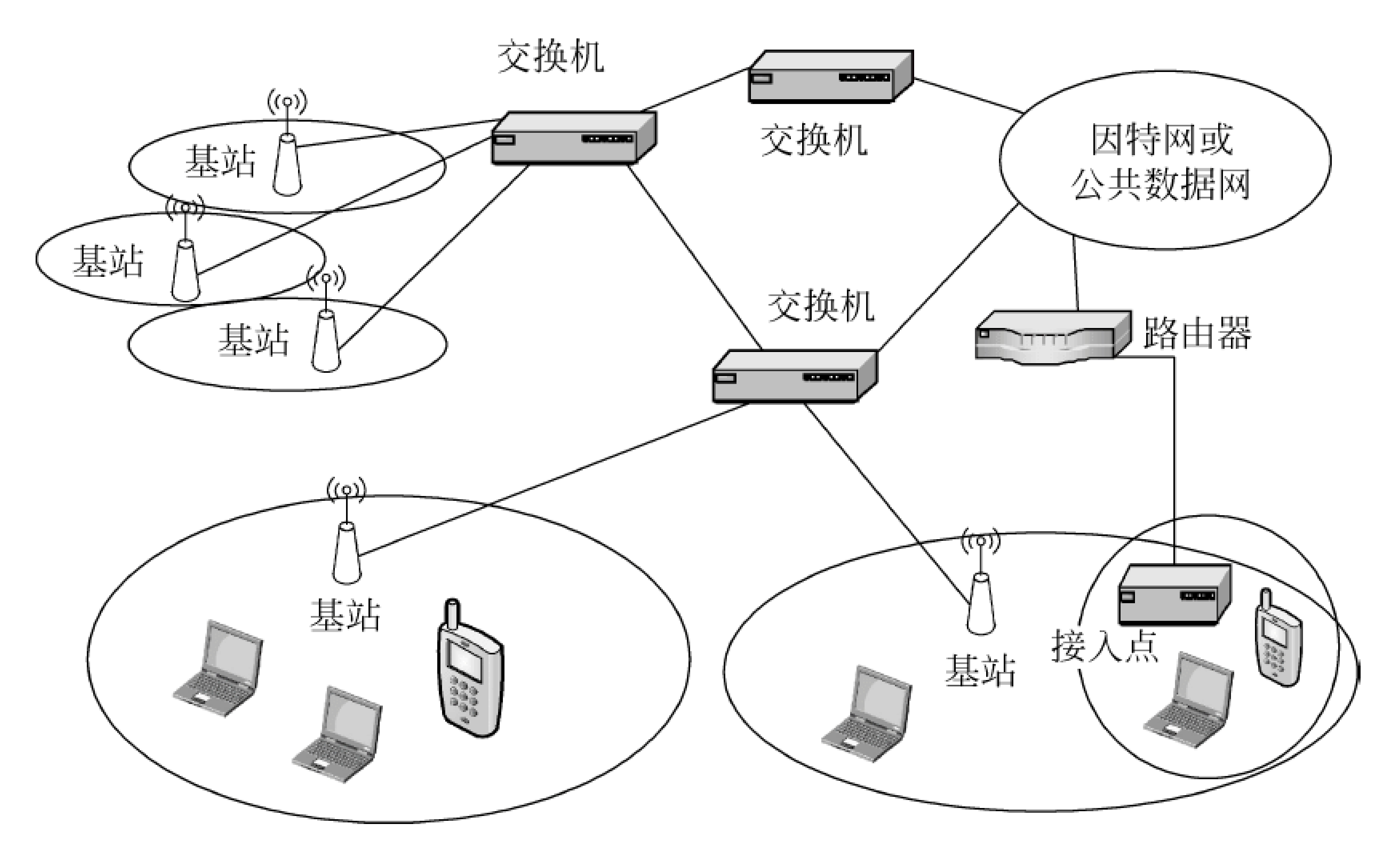


图9.1 无线网络体系结构

node)将进入和离开蜂窝,在不同点接入网络。接入网络技术和连通与断开的模式将变化很大,但还必须是无缝进行的。水平交接必须是在同样技术的两点间进行,而垂直交接是在不同类型的两点之间进行。

无线网络可以分为无线广域网、无线城域网、无线局域网和无线个人网络。各种无线网络使用的传输介质及其传输距离如图 9.2 所示。

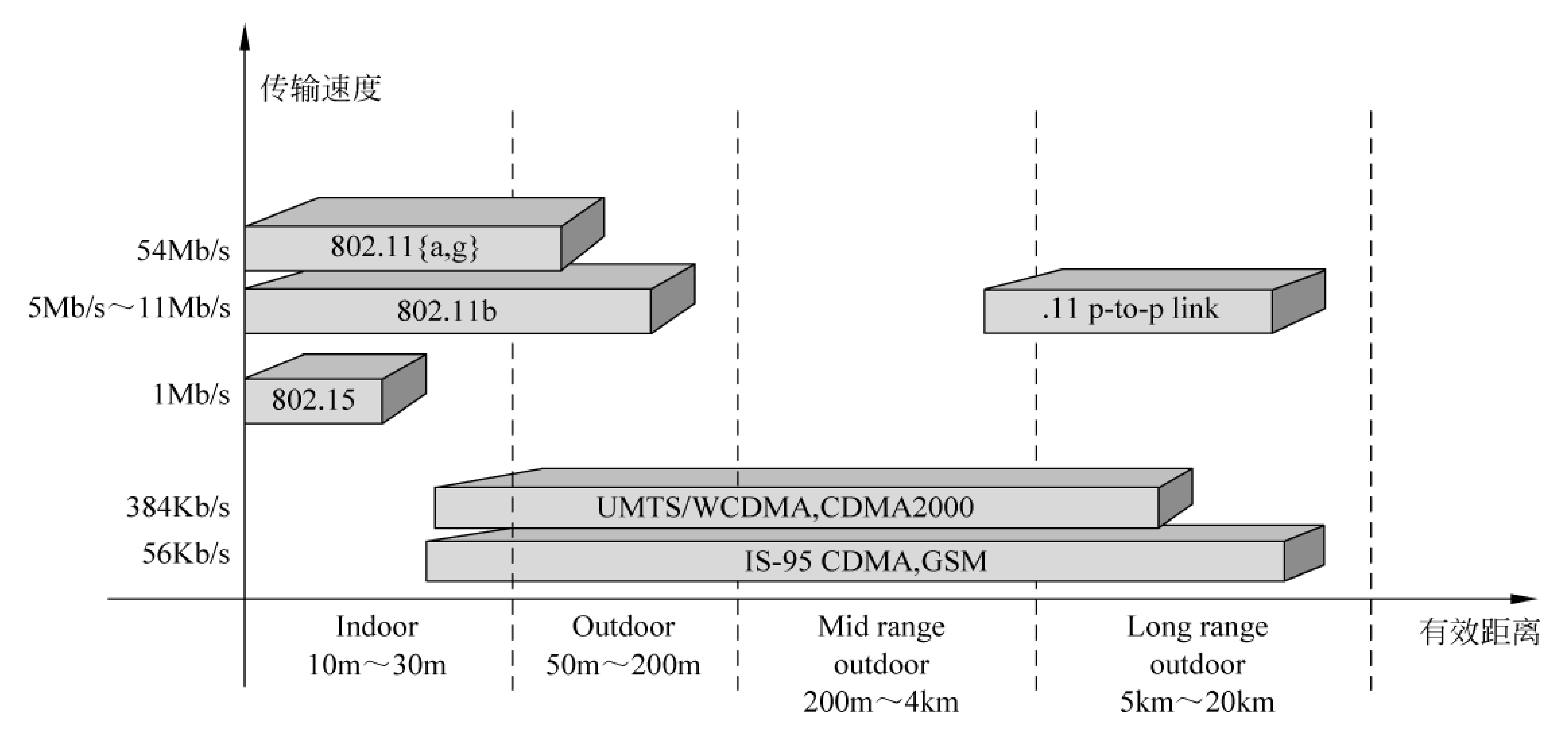


图 9.2　无线网络家谱

(1) 无线广域网(wireless wide area network,WWAN)

主要是指通过移动通信卫星进行的数据通信,其覆盖范围最大。代表技术有 3G,以及未来的 4G 等。由于 3G 的标准化工作日趋成熟,一些国际标准化组织(如 TIU)重点关注能提供更高无线传输速率和灵活统一的 IP 网络平台的下一代移动通信系统,一般称为后 3G、增强型 IMT-2000(enhanced IMT-2000)、后 IMT-2000(system beyond IMT-2000)或 4G。

(2) 无线城域网(wireless metropolitan area network,WMAN)

主要是通过移动电话或车载装置进行的移动数据通信,可以覆盖城市中的大部分区域。其代表技术是 2002 年提出的 IEEE 802.20,主要研究移动宽带无线接入(mobile broadband wireless access,MBWA)技术和相关标准的制定。该标准更加强调移动性,它是由 IEEE 802.16 的宽带无线接入(broadband wireless access,BBWA)技术发展而来的。

(3) 无线局域网(wireless local area network,WLAN)

一般用于区域间的无线通信,其覆盖范围较小。代表技术是 IEEE 802.11 系列。数据传输速率在 11Mb/s～56Mb/s 之间,甚至更高。

无线局域网采用行业标准 802.11。多数 802.11 设备都经过 Wi-Fi 联盟认证,以确保无线局域网产品的互操作性。

公认的 802.11 标准有三种:802.11b、802.11g 和 802.11a。802.11b 是最常用的一种

802.11 标准,被广泛部署在企业和公共区域。传输速率最高 11Mb/s,使用 2.4 GHz 无线频段。802.11g 兼容 802.11b,传输速率最高可达 54 Mb/s,使用 2.4 GHz 无线频段。802.11a 传输速率最高可达 54Mb/s,使用 5 GHz 无线频段。此外,新兴 802.11n 标准较前代协议,提供更高数据传输率和更远的传输距离。理想条件下,802.11n 设备之间的传输速率可达 270Mb/s,它具有与 802.11a 规格相同的双模功能,兼容前三种标准。

从无线局域网的发展历史来看,最开始出现的是传输速度只有 2Mb/s 的 802.11,由于速度比较慢而且协议不是很成熟,很快就被速度为 11Mb/s 的 802.11b 取代,随后,更高速的 802.11a 出现。802.11b 采用相对较为简单的"顺序传播频谱"技术,而 802.11a 采用的是"正交频分多路复用"技术。尽管 802.11a 速度快,但是其工作的 5GHz 频段存在很多问题,其中最主要的是兼容性。由于频率不同,802.11a 产品与基于 802.11b 的产品不能实现互操作。

为解决 802.11a 和 802.11b 不兼容问题,IEEE 开发了 802.11g,工作在 2.4GHz 频段,因而实现了与旧式系统的兼容。802.11g 速度可达到 54Mb/s,对障碍物的穿透能力也较强,在性能方面与 802.11a 相似,而且其设备价格比 802.11a 更便宜,并可兼容 802.11b。

(4) 无线个人网(wireless personal area network,WPAN)

由 IEEE 802.15 定义,无线传输距离一般在 10m 左右。2002 年,IEEE 802.15 工作组成立,专门从事 WPAN 标准化工作。它的任务是开发一套适用于短程无线通信的标准,即 WPAN 的 802.15 系列标准。目前,IEEE 802.15 WPAN 共有 4 个工作组。

- 802.15.1:蓝牙 WPAN 工作组。蓝牙是无线个人局域网的先驱,在初始阶段,IEEE 并没有制定蓝牙相关的标准,所以经过一段快速发展时期后,蓝牙很快就有了产品兼容性的问题。现在,IEEE 制定 802.15.1 行业标准来开发能够相互兼容的蓝牙芯片、网络和产品。
- 802.15.2:共存组。为所有工作在 2.4GHz 频带上的无线应用建立一个标准。
- 802.15.3:高数据率 WPAN 工作组。该标准适用于高质量要求的多媒体应用领域。
- 802.15.4:为了满足低功耗、低成本的无线网络要求,IEEE 标准委员会在 2000 年 12 月份正式批准并成立了 802.15.4 工作组,任务是开发一个低数据率的 WPAN (LR-WPAN)标准。它具有复杂度低、成本极少、功耗很小的特点,能在低成本设备(固定、便携或可移动的)之间进行低数据率的传输。

随着宽带无线 IP 网络技术的发展,WLAN 将和 WWAN 以 IP 为核心实现融合,未来的宽带无线 IP 技术将是由宽带接入和分布式网络构成,具有 2Mb/s 以上的数据传输能力,包括宽带无线固定接入、宽带无线局域网、移动宽带系统和交互式广播网络等。

9.1.2 无线网络安全性分析

无线网络的应用扩展了用户的自由度,还具有安装时间短,增加用户或更改网络结构方便、灵活及经济等特点,可以提供无线覆盖范围内的全功能漫游服务等优势。然而,这种自

由也同时带来了新的挑战，包括安全性问题。由于无线网络通过无线电波在空中传输数据，在数据发射机覆盖区域内几乎所有的无线网络用户都能接触到这些数据。只要具有相同接收频率就可能获取所传递的信息。要将无线网络环境中传递的数据仅仅传送给一个目标接收者是不可能的。另一方面，由于无线移动设备在存储能力、计算能力和电源供电时间方面的局限性，使得原来在有线环境下的许多安全方案和安全技术不能直接应用于无线环境，例如防火墙对通过无线电波进行的网络通信起不了作用，任何人在区域范围之内都可以截获和插入数据，计算量大的加密/解密算法不适宜用于移动设备等。因此，需要研究新的适合于无线网络环境的安全理论、安全方法和安全技术。

与有线网络相比，无线网络所面临的安全威胁更加严重。所有常规有线网络中存在的安全威胁和隐患都依然存在于无线网络中；外部人员可以通过无线网络绕过防火墙，对专用网络进行非授权访问；无线网络传输的信息容易被窃取、篡改和插入；无线网络容易受到拒绝服务攻击和干扰；内部员工可以设置无线网卡以端对端模式与外部员工建立连接。此外，无线网络的安全技术相对比较新，安全产品还比较少。以无线局域网为例，移动节点、AP等每一个实体都有可能是攻击对象或攻击者。由于无线网络在移动设备和传输媒介方面的特殊性，使得一些攻击更容易实施，对无线网络安全技术的研究比有线网络的限制更多、难度更大。

无线网络在信息安全方面有着与有线网络不同的特点，具体表现在以下几个方面。

1. 无线网络的开放性导致其更容易受到攻击

无线链路使得网络更容易受到从被动窃听到主动干扰的各种攻击。有线网络的网络连接是相对固定的，具有确定的边界，攻击者必须物理接入网络或经过几道防线，如防火墙和网关，才能进入有线网络。这样通过对接入端口的管理可以有效地控制非法用户的接入。而无线网络则没有一个明确的防御边界，攻击者可能来自任意节点，每个节点必须面对攻击者的直接或间接的攻击。无线网络的这种开放性带来了非法信息截取、未授权信息服务等一系列的信息安全问题。

2. 无线网络的移动性使得安全管理难度更大

有线网络的用户终端与接入设备之间通过线缆连接，终端不能在大范围内移动，对用户的管理比较容易。而无线网络终端不仅可以在较大范围内移动，而且还可以跨区域漫游，这意味着移动节点没有足够的物理防护，从而易被窃听、破坏和劫持。攻击者可能在任何位置通过移动设备实施攻击，而在全球范围内跟踪一个特定的移动节点是很难做到的。另一方面，通过网络内部已经被入侵的节点实施攻击而造成的破坏更大，更难检测。因此，对无线网络移动终端的管理要困难得多，无线网络的移动性带来了新的安全管理问题，移动节点及其体系结构的安全性更加脆弱。

3. 无线网络动态变化的拓扑结构使得安全方案的实施难度更大

有线网络具有固定的拓扑结构，安全技术和方案容易实现。而在无线网络环境中，动态的、变化的拓扑结构，缺乏集中管理机制，使得安全技术更加复杂。另一方面，无线网络环境中做出的许多决策是分散的，而许多网络算法必须依赖所有节点的共同参与和协作。缺乏集中管理机制意味着攻击者可能利用这一弱点实施新的攻击来破坏系统。

4. 无线网络传输信号的不稳定性带来无线通信网络的鲁棒性问题

有线网络的传输环境是确定的，信号质量稳定，而无线网络随着用户的移动其信道特性是变化的，会受到干扰、衰落和多径等多方面的影响，造成信号质量波动较大，甚至无法进行通信。因此，无线网络传输信道的不稳定性带来了无线通信网络的鲁棒性问题。

此外，移动计算引入了新的计算和通信行为，这些行为在固定或有线网络中很少出现。例如，移动用户通信能力不足，其原因是链路速度慢、带宽有限、成本较高、电池能量有限等，而无连接操作和依靠地址运行的情况只出现在移动无线环境中。因此，有线网络中的安全措施不能对付基于这些新的应用而产生的攻击。

总之，无线网络的脆弱性是由于其媒体的开放性、终端的移动性、动态变化的网络拓扑结构、协作算法、缺乏集中监视和管理点，以及没有明确的防线造成的。因此，在无线网络环境中，在设计实现一个完善的无线网络系统时，除了考虑在无线传输信道上提供完善的移动环境下的多业务服务平台外，还必须考虑其安全方案的设计，这包括用户接入控制设计、用户身份认证方案设计、用户证书管理系统的设计、密钥协商及密钥管理方案的设计等。其中保密性和认证技术是关键。

无线网络环境中安全威胁的具体表现主要如下。

- 攻击者伪装成合法用户，通过无线接入非法访问网络资源。
- 无线链路上传输的未被加密的数据被攻击者截获。
- 针对无线连接或设备实施拒绝服务攻击。
- 不适当的数据同步可能破坏数据的完整性。
- 恶意实体可能得到合法用户的隐私。
- 手持设备容易丢失，从而泄露敏感信息。
- 设备的不适当配置可能造成数据的泄露。
- 恶意用户可能通过无线网络连接到他想攻击的网络上去实施攻击。
- 恶意用户可能通过无线网络，获得对网络的管理控制权限。

要实现信息的机密性、完整性、可用性及资源的合法使用这 4 个基本安全目标，必须采取相应的安全措施对付 4 种基本安全威胁，即信息泄露、完整性破坏、拒绝服务和非法使用。

在安全威胁中，任何一种威胁的实现都会直接导致基本威胁的产生。在无线网络环境下，安全威胁包括无授权访问、窃听、伪装、篡改、重放、错误路由、删除消息和网络洪泛

(flooding)等。

由于受到以上所提到的安全威胁,从而会导致一定的安全风险,如信息窃取、非授权使用资源、窃取服务和拒绝服务等。导致信息窃取的安全威胁有非授权访问、伪装和窃听。导致非授权使用资源的安全威胁有非授权访问、伪装、篡改信息、重放、重路由或错误路由消息等。导致窃取服务的安全威胁有非授权访问、伪装、篡改信息、否认、重放攻击、重路由、错误路由或删除消息。导致拒绝服务的安全威胁有非授权访问、伪装、破坏资源管理信息、重路由、错误路由或删除消息、网络 flooding 等。

9.2　常用无线局域网安全技术

9.2.1　传统安全措施

无线网络面临的安全性挑战是由于无线网络的自身特性而产生的。在有线网络中,有线设备中传输的数据具备固有的安全性。潜在的攻击者必须通过有线连接至电缆设备并受到其他安全手段的防范。当网络中没有连线时,网络用户获得的自由同样扩展到了潜在的攻击者手中,无线网络的安全性受到严重威胁。为此,IEEE 802.11b 从一开始就已经提供了一些基本的安全保障机制来降低潜在安全威胁带来的影响。

无线局域网的开放性要求对无线主机和接入点(或者无线路由器)进行身份认证,防止非法的用户及非法的接入点,以及保障传输数据的完整性和保密性。针对这些要求,IEEE 802.11 提出了服务集 ID(service set identifier,SSID)、MAC 层认证和有线等效保密协议(wire equivalent privacy,WEP),以及 MAC 地址过滤技术等基本安全机制,用来保障无线局域网的数据传输安全。

IEEE 802.11b 可以利用设置无线终端访问的服务集 ID 来限制非法接入。每个 AP 上设置一个服务区域认证 ID,当无线终端设备要连接 AP 时,AP 会检查其 SSID 是否与自己的 ID 一致,只有当 AP 和无线终端的 SSID 相匹配时,AP 才接受无线终端的访问并提供网络服务,如果不符就拒绝给予服务。利用 SSID 可以很好地进行用户群体分组,避免任意漫游带来的安全和访问性能的问题。SSID 提供了一个标志无线局域网边界的方法,所有 SSID 相同的无线设备处于一个无线网络内。因此不知道 SSID 是无法访问特定的无线局域网的。SSID 是一种简单的安全认证接入方法,基本的原理就是把允许入网的标识发送给提供服务的网络,实现网络对终端的认证。

802.11 标准提供一种简单的 MAC 层的安全认证机制,包括两种认证模式:开放系统认证(open system)和共享密钥认证(shared secret)。这两种认证模式是从鉴别无线站点或设备的角度出发,而并未实施对用户身份的认证。在基础模式结构的 WLAN 中认证在无线站点和 AP 之间进行,在 Ad-Hoc 模式结构的 WLAN 中认证在两个无线站点间进行。

802.11 客户端站点的认证过程如图 9.3 所示。

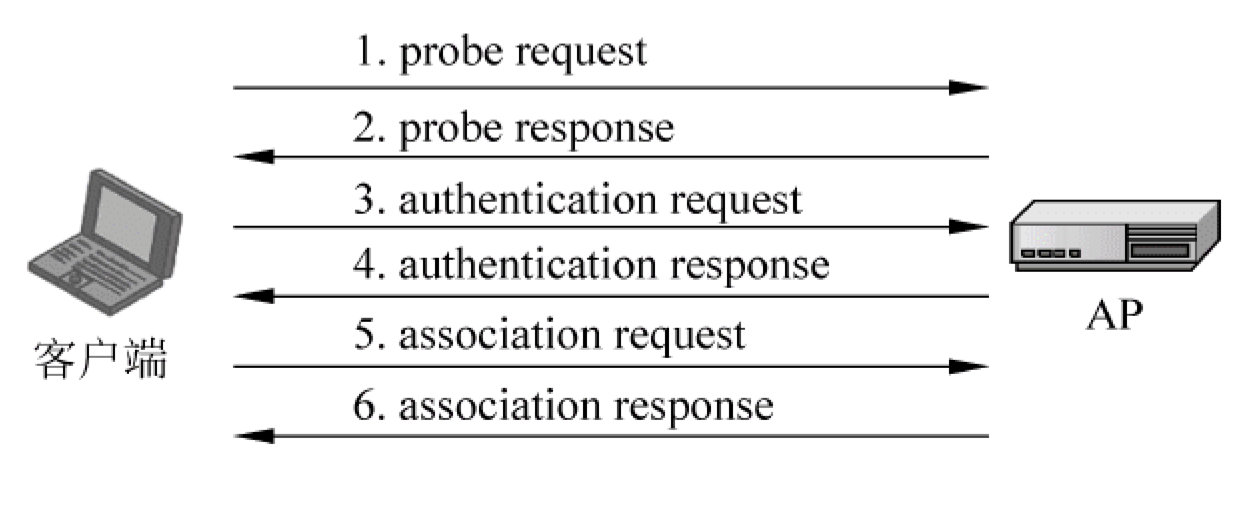


图 9.3 802.11 认证过程

① 无线客户端在每个信道广播探测请求帧(probe request),寻找可以接入的 AP。

② 通信范围内的接入点以探测响应帧(probe response)应答。

③ 无线客户端选择信号最强的接入点,并发送认证请求。

④ 接入点发送认证应答。

⑤ 如果认证成功,无线客户端向接入点发送联合请求(association request)帧。

⑥ 接入点以联合响应(association response)帧应答。

联合成功后,无线客户端可以通过接入点访问网络。

开放系统认证方式非常简单,本质上是一个空的认证过程。任何想接入网络的无线主机都可以通过该认证过程,只要无线主机的认证模式也设为 OpenSystem,并且 SSID 正确就都可通过 AP 访问网络。802.11 的默认认证方式是开放系统认证方式。

共享密钥认证采用 WEP 实现。WEP 是所有经过 Wi-Fi 认证的无线局域网所支持的一项标准功能,是 IEEE 802.11b 协议中最基本的无线安全加密措施,是 802.11 标准中用来保护无线传输过程中的链路级数据的协议,其主要用途是提供接入控制,防止未授权用户访问网络。WEP 使用对称加密算法对数据进行加密,防止数据被攻击者窃听、篡改或伪造。

WEP 依赖通信双方共享一个秘密密钥保护传输帧中的用户数据。加密的过程非常简单:使用简单的密钥和初始化向量按一定的规则组成密钥序列去异或预加密的明文和明文的 CRC 校验组合。

WEP 协议提供如下安全服务。

- 保密性。
- 访问控制:802.11 标准中包含了丢弃不正确的 WEP 加密帧的选择,达到访问控制的功能。
- 数据完整性:采用 CRC 校验防止传输信息被篡改。

WEP 利用对称加密方案,在数据的加密和解密过程中使用相同的密钥和算法,通过防止无 WEP 密钥的非法用户获得网络的访问权来实现访问控制的目的,并通过只允许有正确 WEP 密钥的用户对 WLAN 的数据进行加密和解密来达到数据保密的目的。WEP 加密采用静态的保密密钥,它利用一套基于 40 位共享加密秘钥的 RC4 加密算法对网络中所有通过无线传送的数据进行加密,从而有效地保护数据的传输。各 WLAN 终端使用相同的

密钥访问无线网络。当加密机制功能启用，客户端要尝试连接上网时，即会发出一个“挑战”消息(challenge packet)给客户端，客户端利用共享密钥将此值加密后送回 AP 以进行认证比对，结果正确才能接入无线网络。

WEP 协议实现的关键是 40 位的共享加密密钥的分发、RC4 加密算法的实现和 CHAP 验证算法的实现。

WEP 的缺点是缺少动态密钥管理及加密自身的安全性不足，后者可随时间的推移将密钥暴露给攻击者。

另一种方法是 MAC 地址过滤技术。它通过检查用户数据包的源 MAC 地址来认证用户的可信度，即限制接入终端的 MAC 地址，以确保只有经过注册的用户才可以接入无线网络。由于每一块无线网卡拥有唯一的 MAC 地址，在 AP 上可以建立一张“MAC 地址控制表”，只有地址匹配时 AP 才允许无线终端接入。MAC 地址控制可以有效防止未经过授权的用户侵入无线网络，但却增加了网络管理的复杂性，同时也给用户接入网络带来了不便，因为新增的用户无法便利地接入网络，并且用户更换网卡或无线终端设备后需要重新接入网络。

MAC 地址过滤技术并不是 IEEE 802.11 中规定的，但是很多设备供应商都支持这一用户认证技术，即为无线局域网的每一个接入点设置许可接入的用户的 MAC 地址清单。MAC 地址不在清单中的用户，接入点将拒绝其接入请求。另一种情况是，MAC 地址清单集中放置在后台认证服务器内(如 RADIUS 服务器)，每当用户申请接入时，接入点将接入请求转发给认证服务器。认证服务器通过比对允许接入的地址列表，返回允许接入或拒绝接入的消息。

尽管 802.11 标准提供了以上一些基本的安全机制抵御潜在的安全威胁，但其提供的安全性有限和安全保障能力有限，这是因为如下原因。

① SSID 可以用来区别不同的无线局域网，但单凭 SSID 无法防止非法用户的接入。因为 SSID 是预先设置的，保密性不高。而且 802.11 协议规定的一种用户接入方式是无线用户监听信道接收 AP 定时广播的 SSID，以便于用户找到正确的网络。这种方式也使得其他用户可以轻易地发现可用的 SSID，使其没有 SSID 仍可以找到网络。即使不进行 SSID 广播，攻击者同样可以通过网络嗅探和数据包分析等方法得到该网络的 SSID。因此，仅依靠 SSID 进行用户认证不能阻止非法入侵者接入无线网络。

② MAC 地址过滤机制基于 WLAN 网卡都有一个出厂时设定的 48 位 MAC 地址。只需在每个 AP 保存一份合法 MAC 地址列表，只有在列表中的设备才能接入网络。问题在于合法 MAC 地址列表必须及时更新。一旦有用户的 MAC 地址改变，管理员就必须更新所有的地址列表以适应用户的变化。在拥有成百台设备的企业应用环境中，维护列表数据有很大的工作量。同时，MAC 地址过滤并不是绝对安全。攻击者可以通过监听无线通信，从用户的数据中得到合法的 MAC 地址，并使用合法的 MAC 地址来实现入侵。因此，MAC 地址过滤适用于规模较小、安全级别不是很高的网络。

③ 共享密钥认证。由于使用 WEP 加密算法，其认证过程给攻击者提供了轻松获得共享密钥的途径：在共享密钥认证过程中采用明文方式传输质询文本；并且传输经过 WEP 加密的质询文本。攻击者通过截取这两段数据，经过简单的 XOR 运算就可得到用于 WEP 加密和解密的密钥流，只要再获得 IV，攻击者就可以通过 RC4 算法推算出 WEP 加密使用的共享密钥。由 WEP 算法可看出，虽然 IV 的存在延长了密钥共享密钥的使用寿命，因为(Key，IV)对随着 IV 的变化而变化，增加了窃取密钥的难度，可以使共享密钥不用频繁更换。但 IV 是 24 位的，那么 RC4 可以使用的 IV 就是有限的。如此可见，IV 的重用不可避免。如果攻击者知道两个用相同 IV 加密的信息包之一的明文，就可以对这两个信息包进行解密。在 IV 的选择中，有些数字在 RC4 加密算法中效果不好，被称作 Weak IV。使用低强度 IV 加密的数据包很容易经过一定的运算得到部分 WEP 密钥。因此，通过大量监听数据，攻击者可以得到足够的 Weak IV，从而破解出 WEP 密钥，危及网络安全。

④ 802.11 局域网中 WEP 密钥的管理也可能影响到网络的安全性。802.11 标准提供了使用 WEP 密钥的两种方案：一种是提供一个包含 4 个密钥的窗口。无线站点或 AP 可以用这 4 个密钥中的任何一个来加密或解密数据包，但传输时只能使用这 4 个人工输入的密钥中的一个，即默认密钥。第二种方法叫密钥映射表。在这种方法中，每一个唯一的 MAC 地址可以拥有一个独立的密钥，使用每一用户的独立密钥使其他人很难针对密钥进行攻击。但是，由于密钥只能手工改变，实施合理的密钥管理成为一个重要和困难的问题。

根据以上的分析可知，由于无线网络的固有特性和 802.11 在安全管理方面的欠缺，无线局域网没有可靠的安全保障，可能受到多种类型的网络黑客的攻击，如欺诈访问点(合法的移动用户可能会通过非法的访问点接入网络，造成严重后果)。未通过授权就使用网络资源也是对无线网络安全的威胁之一，采用良好的认证机制(如 802.1x)可以解决这个问题。此外，还存在拒绝服务攻击、MAC 地址欺骗、会话劫持、流量分析及偷听等威胁。

9.2.2 增强安全机制

1. 802.1x 认证机制

802.1x 是基于 IEEE 标准的网络认证框架。它不仅限于无线网络，还可以用于高端有线 LAN 设备上。802.1x 利用 RADIUS(远程身份验证拨入用户服务)网络身份验证和授权服务验证客户端的身份。802.1x 使用 EAP 解决不同组件间的身份验证问题，并生成保护客户端与网络访问的密钥。

802.1x 要求无线工作站安装 802.1x 客户端软件，无线访问点要内嵌 802.1x 认证代理，同时它还作为 RADIUS 客户端，将用户的认证信息转发给 RADIUS 服务器。802.1x 除提供端口访问控制能力之外，还提供基于用户的认证及计费，特别适合于公共无线接入解决方案。

IEEE 802.1x 本身并不提供实际的认证机制，802.1x 引入 PPP 协议定义的扩展认证协议 EAP。本质上，802.1x 的安全接入是建立在已有的认证技术的基础上引入端口控制的

概念。EAP 可用于多种基于密码、公钥证书或其他凭据的不同身份验证方法。因为 EAP 是一种可插入身份验证方法，因此有多种不同的 EAP 类型。最佳的 EAP 类型实质上使用加密来保护身份验证会话，并能在认证过程中动态生成用于加密的密钥。不同的基于 802.1x 的 WLAN 安全解决方案提供不同的 EAP 类型及不同级别的保护。这些解决方案在各操作系统和网络硬件供应商中有不同的支持级别。EAP、基于 802.1x 的身份验证和网络访问只是构成安全认证解决方案的一部分。

2. IEEE 802.11i 与 WPA

WEP 协议在终端接入和数据传输两方面都采取了安全措施，但是 WEP 采用的是静态密钥，而且同一个服务区内使用的是同一个密钥，存在很大的安全隐患。由于 WEP 缺少足够的安全性，从而延缓了无线局域网在许多企业中的广泛采用。为了帮助提高无线局域网的安全性，IEEE 802.11 工作组成立了 802.11i 任务组，为 802.11 标准开发安全升级。802.11i 任务小组围绕基于 802.1x 端口认证为用户和设备认证开发 802.11i 标准，包括两项重要内容：Wi-Fi 保护接入 WPA（Wi-Fi Protected Access）和强健的安全网络（robust security network，RSN）。

WPA 是 Wi-Fi 联盟提出的统一和改进无线网络安全的策略。WPA 集合了一套安全功能，它们在一定程度上增强了 WLAN 的安全性。WPA 支持强健的加密，从而使发现加密密钥更为困难。

IEEE 802.11i 采用动态密钥，当一台接入点设备与无线客户端设备完成第一次会话后，能够自动生成下一次会话所需的 128 位加密密钥。这样就保证了每个网络用户和每次网络会话所使用的密钥都是唯一的，而且是动态分配的。

IEEE 802.11i 标准草案中主要包含加密技术：TKIP（temporal key integrity protocol）和 AES（advanced encryption standard）。

- TKIP：新一代的加密技术 TKIP 与 WEP 一样基于 RC4 加密算法，且对现有的 WEP 进行了改进，在现有的 WEP 加密引擎中增加了“密钥细分”（每发一个包重新生成一个新的密钥）、“消息完整性检查”、“具有序列功能的初始向量”和“密钥生成和定期更新”4 种算法，增强了加密安全强度。
- AES：IEEE 802.11i 中还定义了一种基于“高级加密标准”AES 的加密算法，以实施更强大的加密和消息完整性检查。AES 是一种对称的块加密技术，提供比 WEP/TKIP 中 RC4 算法更高的加密性能，为无线网络带来更强大的安全防护。WPA 相对 WEP 来说在身份认证算法上作了改进并增强了加密算法。系统的关键是动态密钥和加密算法的更新。

其中，TKIP 是 WPA 中为了改进 WEP 的安全性而使用的加密协议和算法。它改变了密钥生成方式，通过更频繁地变换密钥来获得安全，还增加了消息完整性检查功能来防止数据包伪造。

RSN 是接入点与移动设备之间的动态协商认证和加密算法。802.11i 草案标准中建议的认证方案是基于 802.1x 和扩展认证协议 EAP 的，加密算法则为高级加密标准 AES。动态协商认证和加密算法使 RSN 可以不断演进，与最新的安全水平保持同步，增加算法降低新的安全威胁，并不断提供保护无线局域网传输的信息所需要的安全性。

由于采用动态协商、802.1x、EAP 和 AES，RSN 比 WEP 和 WPA 可靠得多。但 RSN 不能很好地在遗留设备上运行，只有最新的设备才拥有加快算法在客户机和接入点中运行所需的硬件，提供无线局域网产品所期望的高性能。

WPA 可以把遗留设备的安全性提高到可接受水平，而 RSN 则是提高 802.11 无线传输安全性的更高级的技术。目前，最新的 Microsoft 产品支持 WPA，并提供 WEP 和动态密钥一起使用的方法，从而提高 WLAN 的安全性。WPA 和动态 WEP 选项都支持使用 802.1x 和 EAP 来提供基于密码或基于证书的身份验证。

3. WiMAX 安全机制

在 802.16D3 版本中，主要是通过在 MAC 层中定义了一个保密子层来提供安全保障的。保密子层主要包括两个协议：数据加密封装协议和密钥管理协议。其中，数据加密封装协议定义了 IEEE 802.16 支持的加密套件，即数据加密与完整性验证算法，以及对 MAC 帧应用这些算法的规则。而密钥管理协议定义了从基站向用户工作站分发密钥数据的安全方式，两者之间密钥数据的同步及对接入网络服务的限制。802.16D3 版本安全机制的主要工作流程如下。

① 工作站向基站发送一个认证信息消息。该消息包含厂商的 X.509 证书。

② 工作站向基站发送授权请求消息。该消息包括生产商发布的 X.509 证书，基站所支持的加密算法及基站的基本连接 ID。

③ 基站验证工作站的身份，决定加密算法，并为工作站激活一个授权密钥(authorization key，AK)。

④ 基站将 AK 用工作站的公钥加密后返回给工作站。

⑤ 工作站定时发送授权请求消息给基站来更新 AK。

另外，随着 AK 的交换，基站建立了工作站的身份认证及工作站授权接入的服务。亦即在基站和工作站之间建立了某种安全关联。安全关联是基站和一个或多个工作站间共享的一组安全信息，目的是为了支持 IEEE 802.16 网络间的安全通信。安全关联可以包括用来加密数据流的密钥和初始化向量。故在获得授权后，工作站应该向基站请求加密密钥 TEK，流程如下。

① 工作站向基站发送加密密钥请求消息。

② 基站在收到该消息后，生成 TEK，并通过密钥回应消息发送给工作站。

③ 工作站定时发送密钥请求消息给基站更新 TEK。

802.16D3 版本安全机制采用的是单向认证，而且认证机制缺乏扩展性。认证机制只是

基于 X.509 证书，因此缺乏扩展性，没有抗重放攻击保护。另外，在 802.16D3 版本对数据的加密采用的是 DES-CBS 算法，这种算法的密钥长度只有 56 位，容易遭受到穷举攻击。

802.16E 对 D3 版本安全机制进行了完善。802.16E 是 802.16 工作组标准，它为了解决 802.16 中原有安全机制存在的问题，引入了基于 EAP 协议的认证机制，以解决单向认证和认证机制缺乏扩展性的问题。同时，它还引入了 AES-CCM 数据加密协议。AES-CCM 是基于 AES 的 CCM 模式，以解决 D3 版本中安全机制缺乏抗重放保护和加密算法本身不安全的问题。

9.3 802.11X 认证机制

802.11 标准采用的认证机制更多着眼于无线局域网的连通性，在验证无线主机或用户的身份方面没有相应的规定。对企业级无线局域网的应用来讲，需要有一种支持集中用户认证的新的认证框架来弥补 802.11 标准本身的不足。

IEEE 802.11 委员会为 802.11 标准提出了称为健壮安全网络(robust security network，RSN)的安全架构。RSN 采用 802.1x 标准提供接入控制、认证和密钥管理。802.1x 标准为 802.11 标准提供了以下的安全管理措施。

- 用户身份认证。
- 动态密钥派生。
- 双向认证。

802.1x 是开放的标准，提供了扩展认证方式，支持使用智能卡(smart cards)、一次性密码(one-time passwords)和基于证书的认证等多种高层方式。

9.3.1 802.1x 框架结构

IEEE 802.1x 规定了三种实体：请求者、认证者和认证服务器。三种实体之间的通信关系和通信协议如图 9.4 所示。

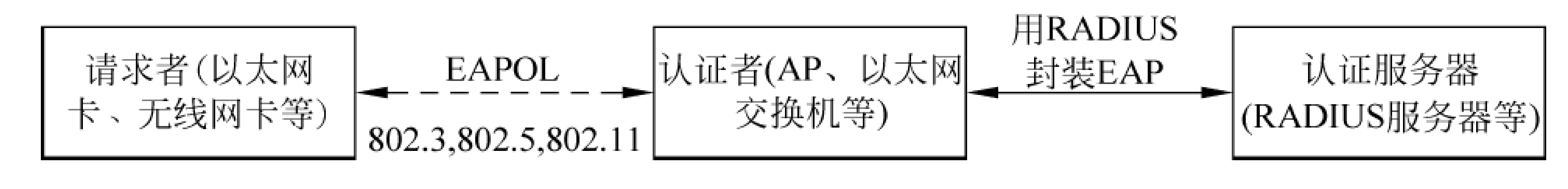


图 9.4 802.1x 的三种实体

请求者实体是指想使用由认证者(如交换机、接入点等)提供的网络服务的实体，即被认证的用户接入设备。认证者一般为接入控制设备，如 AP 或无线路由器等，在接入设备和认证服务器之间转发认证信息，根据认证结果设置端口状态。认证服务器是对请求访问网络

资源的用户设备进行实际认证的设备，常用 RADIUS 认证服务器。认证服务器可以是本地的，即与认证者位于同一设备上；也可以是远程的，通过有线或无线网络与认证者进行通信。

认证系统有两个网络访问端口：不受控端口和受控端口。不受控端口始终保持连接状态，只能通行认证信息；受控端口有“授权”和“未授权”两种状态，可以通行认证信息以外的数据帧。

EAPOL 协议在认证者和请求者间传输 EPA 包，认证服务器和认证者间用 RADIUS 协议通信。EAP 信息是作为 RADIUS 协议的一个属性被传输的，RADIUS 协议是在 AP 和 RADIUS 服务器之间进行每包认证和完整性检验的机制。RADIUS 协议标准要求三个实体的操作遵循特定的状态机顺序。状态机的执行决定包的发送顺序、认证过程的成功或失败及重发时延等。因此，状态机是整个通信安全建立的关键。图 9.5 所示为受控端口和不受控端口的工作方式，认证者实体根据认证结果控制受控端口的状态。如果认证成功，受控端口处于授权状态，用户可以自由访问网络资源。否则，端口处于未授权状态，认证系统拒绝向该用户提供接入服务。

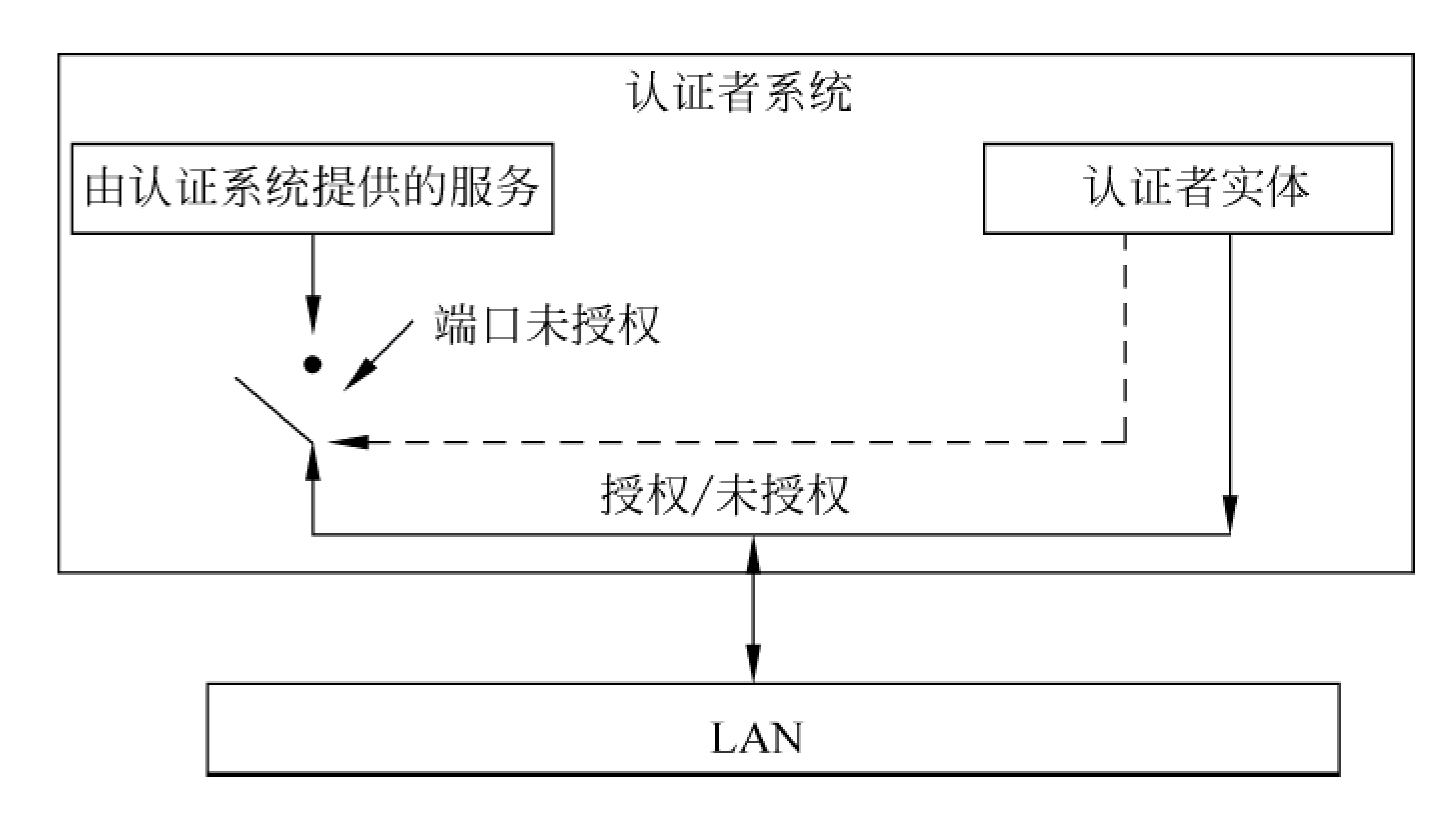


图 9.5 基于端口认证的端口结构

在本无线自组织网实验网络中，802.1x 的三种实体具体由无线主机、无线路由器和 RADIUS 认证服务器组成。

IEEE 802.1x 标准使用可扩展的认证协议，“可扩展”的意思是指任何认证机制可以被封装在 EAP 请求/响应信息包内，因此利用该协议可以实现较广泛的认证机制。EAP 本身是为 PPP 认证制定的一个通用协议，其特点是 EAP 在链路控制(link control protocol，LCP)阶段没有选定一种认证机制，而把这一步推迟到认证阶段。

EAP 是建立在请求/响应通信模型之上的，EAP 工作在网络层上而不是在链路层上。因此，可以将信息转发到中心认证服务器(如 RADIUS)而不是让每一个接入点进行认证，因而有更大的灵活性。在认证成功前，AP 必须允许 EAP 信息通过，为了实现这一点，使用了前面所述的双端口模型。

EAP 并不是一个具体的认证方式，而是一种认证协议的封装格式，通过使用 EAP 封

装，客户端和认证服务器能够实现对具体认证协议的动态协商。

1. EAP 数据包格式

EAP 数据包格式如图 9.6 所示。在该数据包中有 Code(代码)、Identifier(标识符)、Packet Body Length(数据包长度)、EAP Type(EAP 数据包子类型)和 Type-Data(类型-数据)5 个字段域，各域都按照从左到右的顺序在网络中传送。

Code	Identifiter	Packet Body Length
EAP Type	Type Data	

图 9.6　EAP 数据包格式

(1) Code 字段。该字段为一个字节长度，表示 EAP 数据包类型。如果收到的数据包 Code 字段不是有效值，则丢弃该数据包。EAP 数据包的 Code 字段值分配如下。

- EAP-Request：EAP 认证请求。
- EAP-Response：EAP 认证响应。
- EAP-Success：EAP 认证成功。
- EAP-Failure：EAP 认证失败。

(2) Identifier 字段。该字段为一个字节长度，用于匹配请求与应答。认证者系统和 RADIUS 服务器根据 Identifier 字段值可检测出相同请求者系统的重复请求，对重复请求不加处理直接丢弃。

(3) Packet Body Length 字段。该字段为 2 个字节长度，它指的是包含代码、标识符、长度、EAP 数据包子类型和数据域在内的总长度。超出长度域的部分被看作填充字节而被忽略。在请求者系统与认证者系统之间传输的 EAPOL 分组封装 EAP 数据包，其中 EAPOL 的长度字段域值与 EAP 长度域值相同，以后将详述 EAPOL 报文结构。

(4) EAP Type 字段。该字段为一个字节长度，标识 EAP 数据包中 Data 字段的类型。EAP Type 字段值分配如下。

- Identity：用户身份。
- Notification：通知。
- NAK Response：无应答。
- MD5 Challenge：MD5 质询。
- One Time Password(OTP)：一次性密码。
- Generic Token Card(GEN)：通用令牌。

(5) Type Data 字段。该字段域长度不定，不同的 EAP Type 对应不同的 Type Data 值。用户身份的 Type Data 字段域可以是用户名；通知的 Type Data 字段域携带了认证者系统给请求者系统的一段可显示的信息；NAK 类型的数据包指示出请求者系统请求的认

证类型不被认证者系统接受，该类型的数据包仅用于响应数据包中；MD5 质询的 Type Data 字段域是 MD5 的质询文本。

一次性密码类型的请求数据包中包含了 OTP 质询，而与之相对应的响应数据包中包含对 OTP 质询的应答，NAK 类型的应答指示出客户期望的认证机制类型。

2. EAP 协议的交互过程

EAP 认证协议是一个用于 PPP 认证的通用协议，可以支持多种认证方法。EAP 并不在链路建立阶段指定认证方法，而是把这个过程推迟到认证阶段。这样认证者就可以在得到更多的信息以后再决定使用什么认证方法（如智能卡、一次性密码和 kerberos 等多种认证方法）。这种机制还允许 PPP 认证者简单地把收到的认证报文传递给后方的认证服务器，由后方的认证服务器来真正实现各种认证方法。其工作过程如图 9.7 所示。

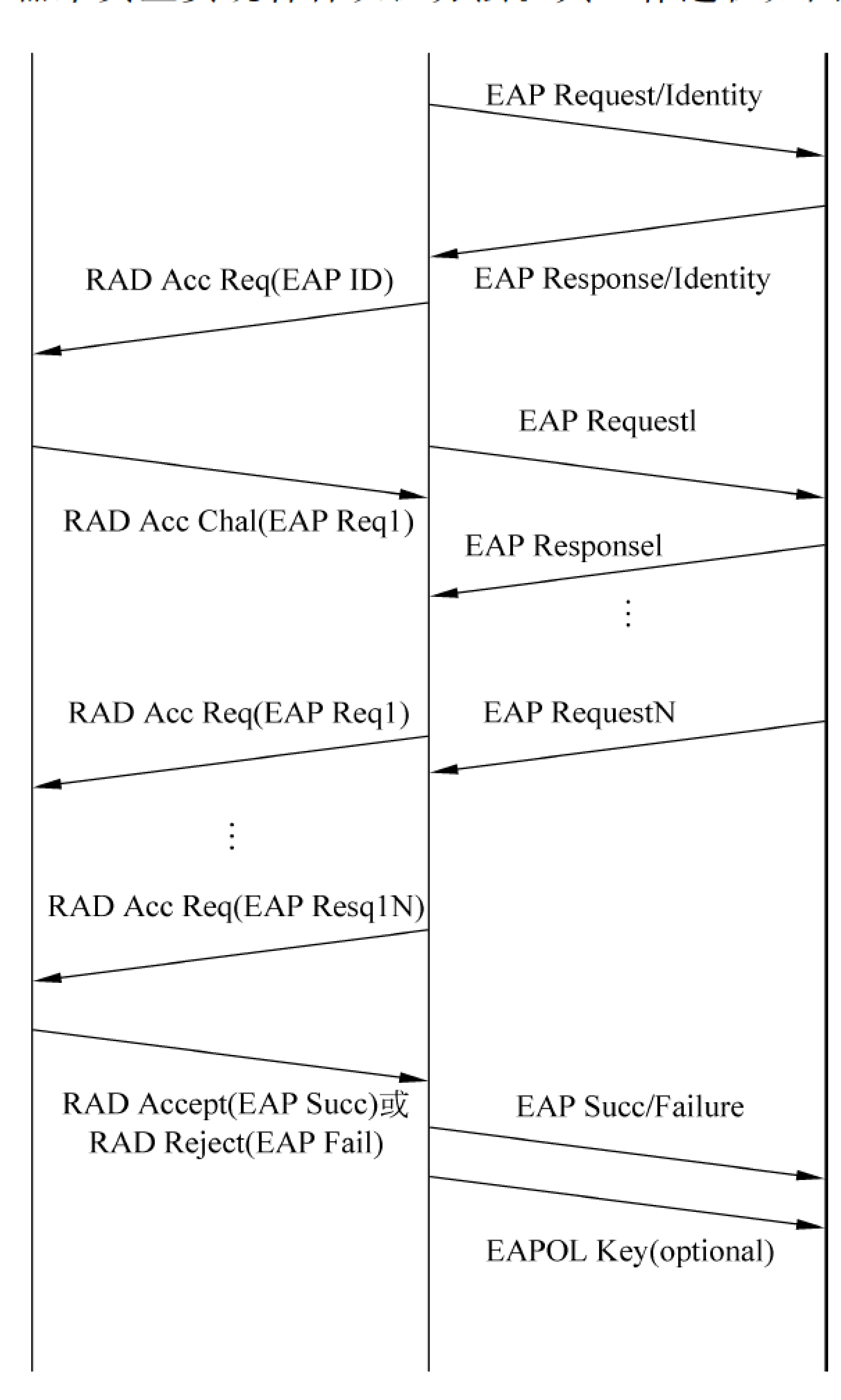


图 9.7　EAP 认证交互过程

在链路阶段完成以后，认证者向请求认证的用户发送一个或多个 EAP 请求报文(EAP request)。在请求报文中通过 EAP 类型字段(EAP type)来指明认证者所请求的信息类型，如请求认证用户的 Identity、MD5 质询文本、一次密码(OTP)及通用令牌卡等。典型情况下，认证者首先发送一个 Identity 请求报文(EAP request/identity)，随后再发送其他的请求报文。当然，并不是必须要首先发送这个 Identity 请求报文，在请求者身份已知的情况下这个步骤可以跳过。

请求认证的用户对每一个请求报文回应一个应答报文。和请求报文类似，应答报文中也包含一个 EAP 类型字段，与请求报文中的类型字段相对应。请求与应答报文成对出现和交换，直到身份和信息的交换过程结束。

最后，认证者通过发送一个成功或者失败的报文结束认证过程。

3. EAPOL 协议

802.1x 标准在认证者系统和请求者系统之间的认证消息传递采用 EAPOL(EAP on line)封装 EAP 数据包。EAPOL 数据包格式如图 9.8 所示，包括 Ethernet Type(以太网类型)、Protocol version(协议版本)、Packet Type(数据包类型)、Packet Body Length(数据包长度)和 Packet Body(数据包体)5 个字段域。

Ethernet Type	Protocol Version	Packet Type
Packet Body Length	Packet Body	

图 9.8　EAPOL 数据包格式

① 以太网类型字段。该字段域为 2 个字节长度，以太网类型值为 0x888E，指示该数据包为 EAPOL 类型。

② 协议版本字段。该字段域为 1 个字节长度，标识支持的 EAPOL 协议版本。当前的 EAPOL 协议版本为 1。

③ 数据包类型字段。该字段域为 1 个字节长度，标识该 EAPOL 数据包的类型。如果收到的 EAPOL 数据包的类型不是有效值，则丢弃该包。数据包类型字段域值的分配如下。

- EAP-Packet：认证信息帧用于承载认证信息。
- EAPOL-Start：认证发起帧。
- EAPOL-Logoff：退出请求帧可主动终止已认证状态。
- EAPOL-Key：密钥信息帧支持对 EAP 报文的加密。
- EAPOL-Encapsulated-ASF-Alert：用于支持 Alert Standard Forum ASF 的 Alerting 消息。

其中，认证发起帧 EAPOL-Start 既可以由请求者发起，也可以由认证者发起。

④ 数据包长度字段。该字段域为 2 个字节长度，它指的是数据包体的长度，如果没有

数据包体则该字段域值为 0。

⑤ 数据包体。该字段长度不定，如果数据包类型为 EAP-Packet、EAPOL-Key 或 EAPOL-Encapsulated-ASF-Alert，数据包体即为相应类型的数据包。而对于其他的数据包类型 EAPOL-Start、EAPOL-Logoff，该字段为空。

9.3.2 802.1x 安全性分析

虽然 IEEE 802.1x 已经提供了更高层次上的安全，但仍然存在不安全的因素，其中最主要的是缺少交互认证过程。协议本身规定了网络对用户身份的认证过程，但忽略了用户判定网络合法性的过程。

这个问题源于状态机中用户和 AP 的异步性，根据 802.1x 标准的规定，仅当认证成功后，受控端口才处于接通状态。但在无线用户端（请求者），其受控端口总是处于认证成功后的接通状态。而认证只是 AP 对无线用户端的单向认证。因为 EAP 协议本身是单向认证的，即无线网络通过 EAP 鉴别无线用户的合法性。

由于 802.1x 的单向认证，使得中间人（man in the middle，MIM）攻击成为可能，如图 9.9 所示。

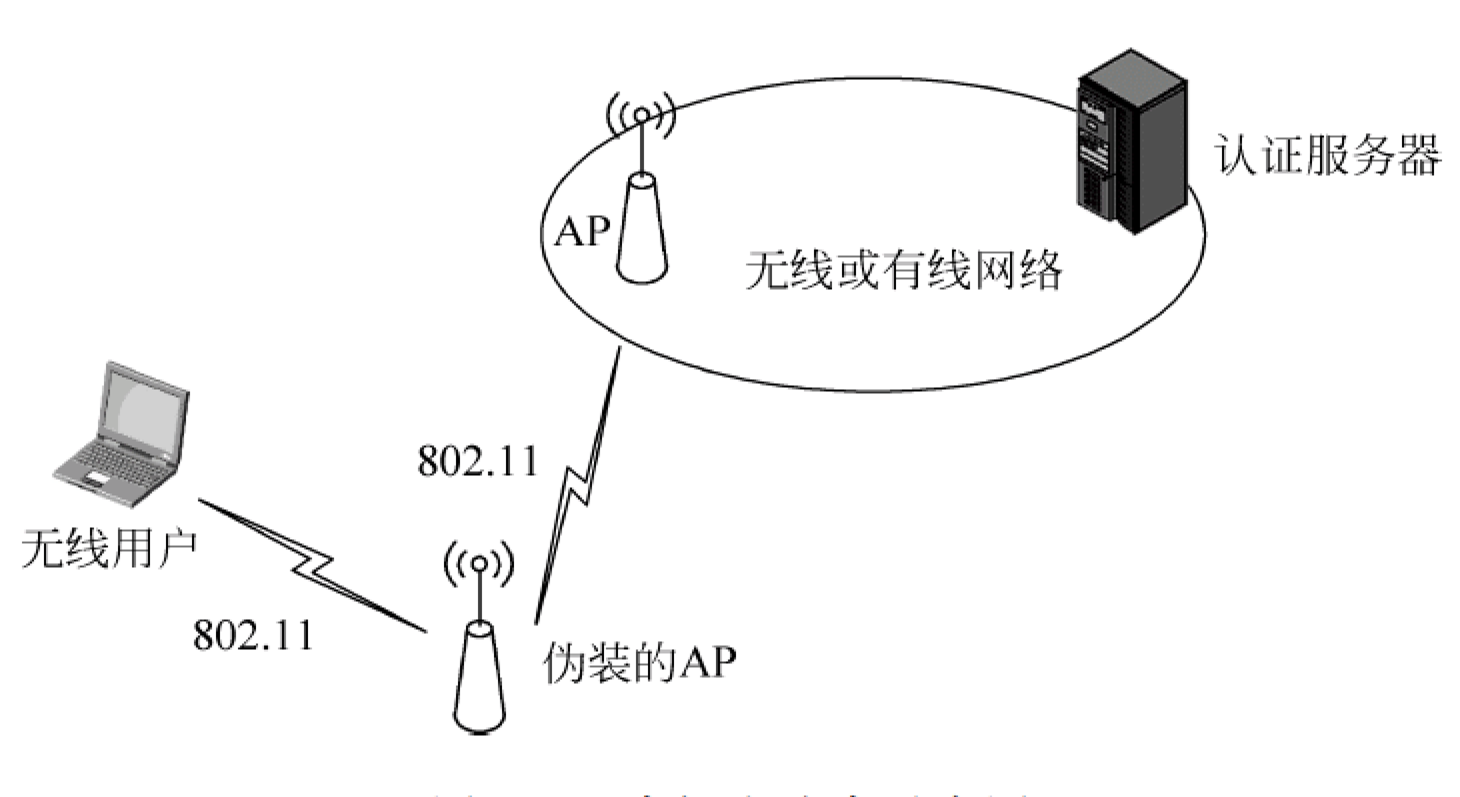


图 9.9 中间人攻击示意图

攻击者处于无线用户和 AP 之间，对无线用户来说，攻击者充当 AP，而对于 AP 来讲它又充当无线用户端。802.1x 认证状态机只接收用户的 EAP 响应并且只向用户发送 EAP 请求信息。类似地，用户状态机不发送任何 EAP 请求信息，很明显，状态机只进行单向认证。这个过程首先就假定了 AP 是完全合法的 AP，在这个假定下的认证框架是不安全的。一个简单的 MIM 如下所述。

当收到来自认证服务器“接受访问（access accept）”的消息后，认证者就向用户发送 EAP Success 信息，向状态机表明认证已成功完成，如果没有使用更高层的认证方法（如 EAP-TLS、EAP-MD5），这一消息并不包含完整性检验信息。在用户状态机中，EAP Success 会触发用户向受控端口传输信息而不用考虑当前状态。简而言之，EAP 成功的信

息会使得用户的受控端口闭合并提供网络连接。因此,攻击者可以以认证者的身份来伪造这一信息,从而有可能开始一个简单的 MIM 攻击。攻击者因此可以获得来自用户的网络通信,使得认证机制不起作用。

另一种攻击方式是会话劫持。当合法用户认证完成后,攻击者窃取 AP 的 MAC 地址向无线用户发送一个 802.11 MAC 解除关联管理帧(disassociation),这使得用户端与 AP 失去关联,从而断开无线连接,而该用户在认证者上的 802.1x 状态机仍处于已认证状态。此时,攻击者可以用已认证的用户端的 MAC 地址获取网络访问,称为会话劫持,如图 9.10 所示。

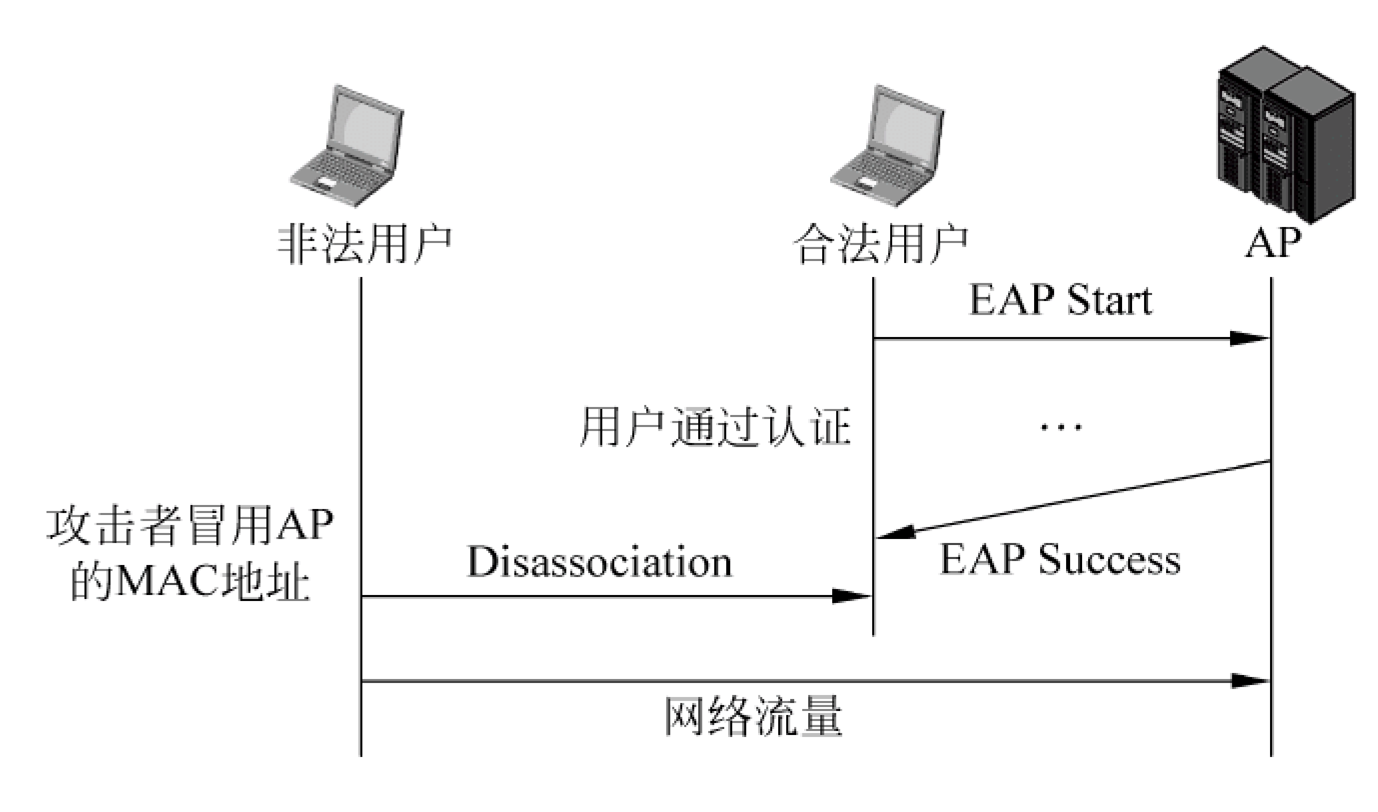


图 9.10　会话劫持示意图

由以上分析可以看出,单向认证缺乏对每个数据包的完整性检验是导致中间人攻击和会话劫持攻击的主要原因。因此,只要在协议实现中对这两个缺陷进行相应的修补,就能使攻击者不能成功。

802.11 与 802.1x 的结合并不能提供健壮的安全无线环境,需要有高层的清晰的交互认证协议来加强。而且 802.1x 也为实现高层认证提供了基本架构,扩展认证协议 EAP 可以支持多种高层认证方式。

9.3.3　高层认证协议

EAP(扩展认证协议)用于在请求者(无线用户)和认证服务器(RADIUS 或其他服务器)之间传递认证信息,实际认证受 EAP 类型限制和控制。充当认证者的接入点只是一个允许请求者和认证服务器进行通信的代理。

目前,IEEE 802.1x 标准可以使用多种 EAP 认证类型,即 RFC 1994 和 RFC 2284 描述的 EAP-MD5；RFC 2716 描述的 EAP-TLS(Microsoft)；基于隧道技术的 EAP-TTLS(Funk)；微软与 Cisco 开发的 PEAP(Protected EAP 保护 EAP)及 Cisco 的 LEAP(Cisco Wireless)。这些认证协议可以使用 RADIUS 服务器统一控制管理,提供动态加密双向认

证,提高无线局域网的安全性。这些认证协议与 802.11 和 802.1x 等协议之间的层次结构如图 9.11 所示。

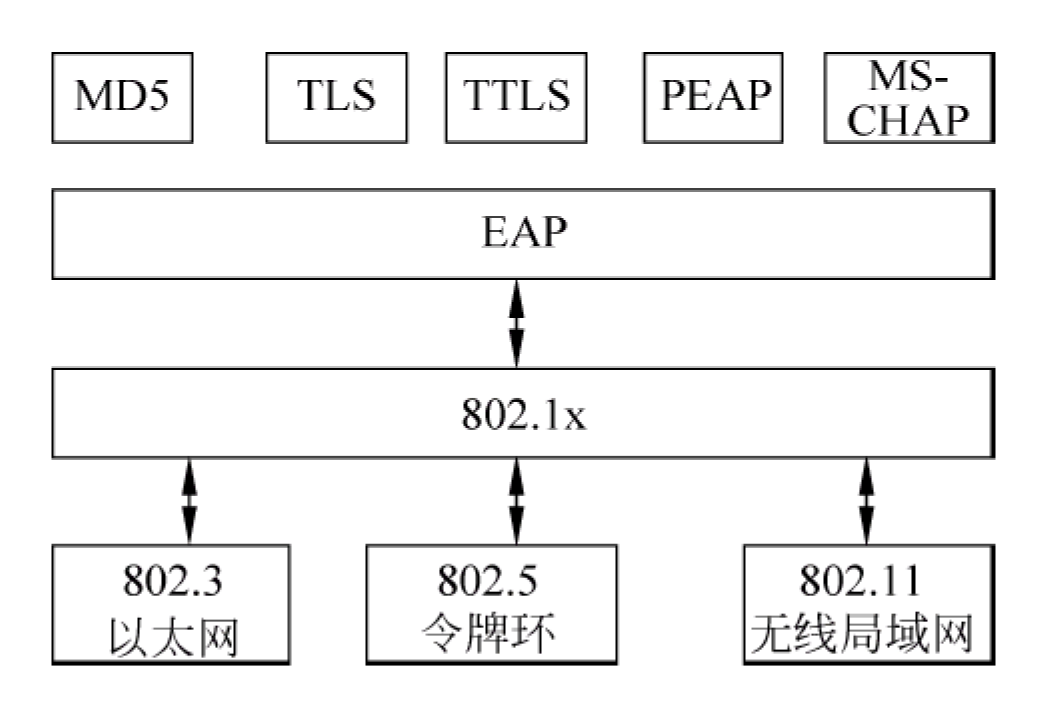


图 9.11 802.1x 及认证协议的层结构

EAP-MD5 认证是一种提供基本级别的 EAP 支持的 EAP 认证类型。对于无线局域网部署,一般建议不要使用 EAP-MD5,因为可以导出用户的密码。

该认证类型通过 Radius 服务器提供简单的集中用户认证。在这种方式下,Radius 服务器不需要证书或者安装在无线工作站中的其他安全信息。用户注册时,Radius 服务器只是检查用户名和密码,如果匹配,就通知无线访问点允许该客户端访问网络服务。

EAP-MD5 仅提供单向验证,只能保证客户端到服务器的认证,不存在无线客户端和网络的相互验证。同时,EAP-MD5 无法根据每个会话动态派生有线对等保密(WEP)密钥。

EAP-TLS(传输层安全)提供了基于证书的验证及无线客户端和网络之间的相互认证,以及动态会话密钥分发。它依赖客户端和服务器端的证书来执行验证,并可用来动态生成基于用户的和基于会话的 WEP 密钥,以保护 WLAN 客户端和接入点之间随后进行的通信。

所有的无线客户端及服务器都需要事先申请一个标准的 X.509 证书并安装,在认证的时候无线客户端和服务器要相互交换证书。在交换证书的同时,客户端和服务器要协商出一个基于会话的密钥,一旦认证通过,服务器将会话密钥传给客户端并通知无线访问点允许该客户端使用网络服务。

EAP-TLS 除了在连接建立时主机和服务器之间分配的会话号(session ID)之外,它需要通过安全连接在客户端和服务器端事先发布认证证书。EAP-TLS 既提供认证,又提供动态会话钥匙分发。RADIUS 服务器需要支持 EAP-TLS 认证和认证证书的管理能力。TLS 支持双向认证,也就是网络(EAP-TLS 服务器)认证终端用户,终端用户认证网络。只有在双向认证通过以后,服务器将向接入认证点发送 EAP-Success 消息,指示用户终端才可以收发数据流。这个消息同时触发了对数据流的加密,在加密密钥建立之前,终端不发送数据。

EAP-TLS 的一个缺点是,必须同时在客户端和服务器端管理证书。对于较大的

WLAN 安装,这可能是一项非常繁重的任务。

目前,除了专门开发的软件外,只有 Windows XP 操作系统支持 EAP-TLS。

EAP-TTLS(隧道传输层安全)是由 Funk Software 和 Certicom 公司开发的,是 EAP-TLS 的一种扩展。该安全方法提供了一种基于证书的认证方法,并通过加密的通道(或“隧道”)进行客户端和网络的相互验证,还提供了一种方法来根据每个用户、每个会话动态派生 WEP 密钥。与 EAP-TLS 不同的是,EAP-TTLS 只需要服务器端的证书。

LEAP(轻量级可扩展验证协议)是一种主要用于 Cisco Aironet WLAN 中的 EAP 验证类型。它使用动态生成的 WEP 密钥对数据传输进行加密,并支持双向验证。Cisco 已允许其他厂商使用 LEAP,因此非 Cisco 适配器也可以使用 LEAP。

PEAP(受保护的可扩展验证协议)提供了一种通过 IEEE 802.11 无线网络安全地传输验证数据的方法,包括传统基于密码的协议。PEAP 通过在 PEAP 客户端和验证服务器之间使用隧道传输来实现此目的。与隧道传输层安全(tunneled transport layer security)这一竞争力的标准相同,PEAP 也仅使用服务器端的证书验证无线 LAN 客户端,因而简化了安全无线 LAN 的实施和管理。Microsoft、Cisco 和 RSA Security 公司共同开发了 PEAP。Cisco 的 LEAP 验证服务器 ACS 最近增加了支持 PEAP 的功能。

RFC 2716 描述了 EAP TLS(transport layer security)认证协议,本协议是基于 RFC 2246 的 TLS 协议制定的,并且获得了微软公司支持的认证算法。TLS 是目前 Web 浏览器使用的安全协议,它的前身是安全套接层(secure socket layer,SSL)。

EAP-TLS 在 SSL v3.0 基础上制定,分为三部分:TLS 握手协议(TLS handshake protocol)、TLS 记录集协议(TLS record protocol)和 TLS 告警协议(TLS alert protocol)。其中,握手协议用来协商 SSL 会话的参数。SSL 的客户和服务器要协商协议版本、加密算法、互相认证和分发密钥。记录集协议帮助 SSL 客户和服务器通过安全隧道交换加密的数据。告警协议在发生错误时通知客户和服务器,另一个功能是会话终结。下面来分析 TLS 的认证过程。

TLS 认证从握手(handshake)开始:首先 SSL 客户端向服务器发送认证请求,服务器将数字证书发送给客户,客户验证收到的证书数字签名的合法性。以上是客户端认证服务器,接下来服务器发送认证请求报文,要求获得客户的身份信息,即客户侧(client-side)认证。客户将自己的数字证书发送给服务器,服务器验证收到证书的合法性。如果以上两次认证都成功,客户和服务器协商加密算法。协商完成后,客户就可以通过加密的隧道传输数据。

图 9.12 中描述了一个完整的 EAP-TLS 交互过程。图中只给出了无线用户与认证者的认证报文交互,实际上无线用户送到认证者的认证报文都被认证者原样打包发送给了服务器,认证实际在无线用户与服务器之间进行。在创建 PPP 链路的 LCP(link control protocol)开始阶段,选择使用 EAP 认证协议;在提交用户身份信息(identity)后开始 EAP-TLS 的用户验证过程。

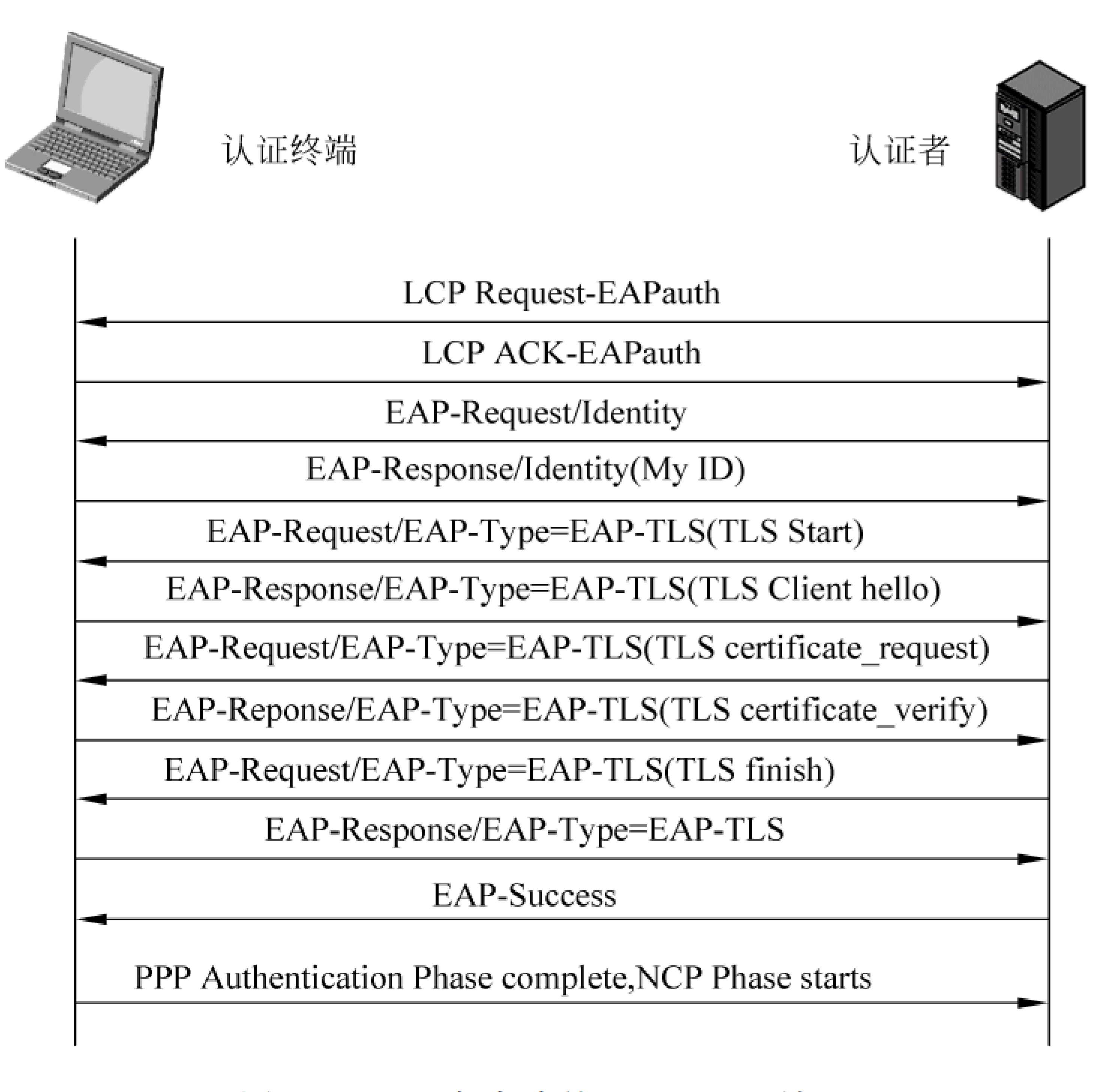


图 9.12 一次成功的 EAP-TLS 认证

客户端向服务器发送 hello 消息，服务器必须回复 hello。两个 hello 消息用于建立安全的会话，hello 中包含协议版本号、会话 ID、密码簇和压缩方法。

hello 还产生两个随机值进行交换(ClientHello. random 和 ServerHello. random)。

密钥交换利用以下 4 个消息：服务器/客户证书、服务器/客户密钥交换(server certificate、server key exchange、client certificate 和 client key exchange)。

如果服务器还未认证，服务器会随同 hello 消息一起发送其证书。有需要时(如服务器无证书)还会发送 server key exchange(服务器密钥交换)消息。

如果服务器已经认证过了，在选择的密码簇支持的情况下，服务器可以向客户要求证书，此时服务器会发送 server hello done 消息表示握手的 hello-message 阶段结束了。然后服务器等待客户的响应。根据服务器的请求，客户发送证书和 client_key_exchange 消息。

然后服务器和客户通过 change cipher spec 消息协商新的算法、密码等。至此，TLS 握手过程结束，服务器和客户可以开始交换应用层数据。

在此过程中通过 TLS 的握手协议完成密钥的协商，详细过程请参考 RFC 2246 The TLS Protocol 及 RFC 2716 PPP EAP TLS Authentication Protocol 文档。

如前所述，EAP-TLS 协议是 802.1x 可以采用的一种高层认证方式，用 EAP 实现对 TLS 的封装。EAP-TLS 除提供 TLS 的认证功能外，还提供分段和重组。

TLS 是一种交互的认证方式，具有完整性保护的密码协商和密钥交换。认证系统对客户

身份进行认证,同时,客户也对认证系统的身份进行认证。EAP-TLS 的报文结构如图 9.13 所示,除了 EAP 头之外,TLS 拥有一些自己的特殊字段,如标志位 Flags 等。

Code	Identifier	Packet Body Length
EAP Type	Flags	TLS Message Length
TLS Message Length		TLS Data

图 9.13 EAP-TLS 数据报结构

9.3.4 802.1x 协议技术特点

1. 协议实现比较简单

① 从网络协议分层角度来看,IEEE 802.1x 协议工作在链路层之上,不需要到达网络层。链路层认证的优势突出,其特点是快速、简单和成本低廉。多数的链路层协议(如 PPP 和 IEEE 802)都可以支持基于链路层的认证技术。客户在认证之前不需要进行服务器的定位,不需要获得 IP 地址。网络接入设备只需要有限的三层功能,就可以轻易实现和 AAA 的结合,从而提供丰富、灵活的认证方式和计费手段。链路层认证处理减小了认证包处理的延时,保证了关键性应用的服务质量。

② IEEE 802.1x 的链路层认证方式可以为上层提供一个平等的认证平台。在多协议网络环境中,基于链路层的认证可以实现对上层应用的完全透明,可以实现和新的网络层协议(如 IPv6)的兼容。高层的认证协议,或者相关的用户数据信息都可以承载在该平台上,而无须考虑下层的传输方式。综上所述,链路层的认证方式的优点是不需要对设备的硬件进行改动,通过软件升级就可以实现新的认证技术引入。并且保留了传统 AAA 认证的网络架构,可以利用现有的 RADIUS 设备。

2. 实现了认证端口和业务端口分离

IEEE 802.1x 的认证体系结构中采用了受控端口和不受控端口的逻辑功能,从而可以实现业务与认证的分离。由 RADIUS 服务器和接入设备利用不受控逻辑端口共同完成对用户的认证与控制,业务报文直接承载在正常的二层报文上通过受控端口进行交换;用户通过认证后,业务流和认证流实现分离。

3. 提供安全可靠的认证技术

IEEE 802.1x 在二层网络上结合 MAC 地址、端口、用户和密码等实现用户认证。在无线局域网网络环境中,IEEE 802.1x 利用 EAP-TLS、EAP-TTLS 等高层认证协议,可以实现用户与网络的双向认证,及对 WEP 证书密钥的动态分配,克服无线局域网接入中的安全漏洞。

4. 应用灵活

IEEE 802.1x 可以灵活控制认证的颗粒度，用于对单个用户连接、用户 ID 或者是对接入设备进行认证。认证的层次可以进行灵活的组合，满足特定的接入技术或者是业务的需要，并且易于运营：控制流和业务流完全分离，易于实现跨平台多业务运营。

综合以上的特点，IEEE 802.1x 标准从制定以来逐渐成为以太网和无线局域网的主要认证协议，几乎所有的主流数据设备厂商在其设备，包括路由器、交换机和无线 AP 上都提供对该协议的支持。在客户端方面，微软 Windows XP 操作系统内置支持，Linux 也提供了对该协议的支持。

9.4 WAPI

随着无线局域网的应用越来越广泛，它的安全问题也日益受到关注，很多组织都在探索新的途径以获得更强的 WLAN 安全保障。2003 年 5 月 12 日，我国发布了无线局域网国家标准 GB 15629.11，本方案已由 ISO/IEC 授权的机构 IEEE Registration Authority（IEEE 注册权威机构）审查并获得认可，分配了用于 WAPI 协议的以太网类型字段，这是目前我国在这一领域唯一获得批准的协议。

该标准的一个重要组成部分就是由宽带无线 IP 标准工作组制定的新的安全机制——无线局域网鉴别和保密基础结构（WLAN authentication and privacy infrastructure，WAPI）。WAPI 主要是针对 IEEE 802.11 中 WEP 协议安全漏洞问题而提出的 WLAN 安全解决方案。这种安全机制由 WAI（WLAN authentication infrastructure）和 WPI（WLAN privacy infrastructure）两部分组成，WAI 和 WPI 分别实现对用户身份的鉴别和对传输数据的加密。WAPI 能为用户的 WLAN 系统提供全面的安全保护。

WAPI 采用国家密码管理委员会办公室批准的公开密钥密码体制的椭圆曲线密码算法和秘密密钥密码体制的分组密码算法，分别用于 WLAN 设备的数字证书、密钥协商和传输数据的加解密，从而实现设备的身份鉴别、链路验证、访问控制和用户信息在无线传输状态下的加密保护。

WAI 采用公开密钥密码体制，利用证书来对 WLAN 系统中的 STA 和 AP 进行认证。WAI 定义了一种名为 ASU（authentication service unit）的实体，用于管理参与信息交换各方所需要的证书（包括证书的产生、颁发、吊销和更新）。证书里面包含有证书颁发者（ASU）的公钥和签名，以及证书持有者的公钥和签名（这里的签名采用的是 WAPI 特有的椭圆曲线数字签名算法），是网络设备的数字身份凭证。

在具体实现中，STA 在关联到 AP 之后，必须相互进行身份鉴别。先由 STA 将自己的证书和当前时间提交给 AP，然后 AP 将 STA 的证书、提交时间和自己的证书一起用自己的

私钥形成签名，并将这个签名连同这三部分一起发给 ASU。

所有的证书鉴别都由 ASU 来完成，当其收到 AP 提交来的鉴别请求之后，会先验证 AP 的签名和证书。当鉴别成功之后，进一步验证 STA 的证书。最后，ASU 将 STA 的鉴别结果信息和 AP 的鉴别结果信息用自己的私钥进行签名，并将这个签名连同这两个结果发回给 AP。

AP 对收到的结果进行签名验证，并得到对 STA 的鉴别结果，根据这一结果来决定是否允许该 STA 接入。同时 AP 需要将 ASU 的验证结果转发给 STA，STA 也要对 ASU 的签名进行验证，并得到 AP 的鉴别结果，根据这一结果来决定是否接入 AP。

由于 WAI 中对 STA 和 AP 进行了双向认证，因此对于采用“假”AP 的攻击方式具有很强的抵御能力。

9.4.1 WAPI 的工作原理

WAPI 的工作原理如图 9.14 所示，整个系统由移动终端(mobile terminal，MT)、接入点(access point，AP)和认证服务器(authentication serrer，AS)组成。其中，认证服务器的主要功能是负责证书的发放、认证与吊销等；移动终端与 AP 上都安装有认证服务器发放的公钥证书，作为自己的数字身份凭证。当移动终端登录至无线接入点时，在访问网络资源之前必须通过 AS 进行双向身份认证。即持有合法证书的移动终端能且只能接入持有合法证书的无线接入点。这样，一方面可以防止非法移动终端接入 AP 未经授权就使用网络资源，另一方面还可以防止移动终端登录至非法 AP 而造成信息泄露。

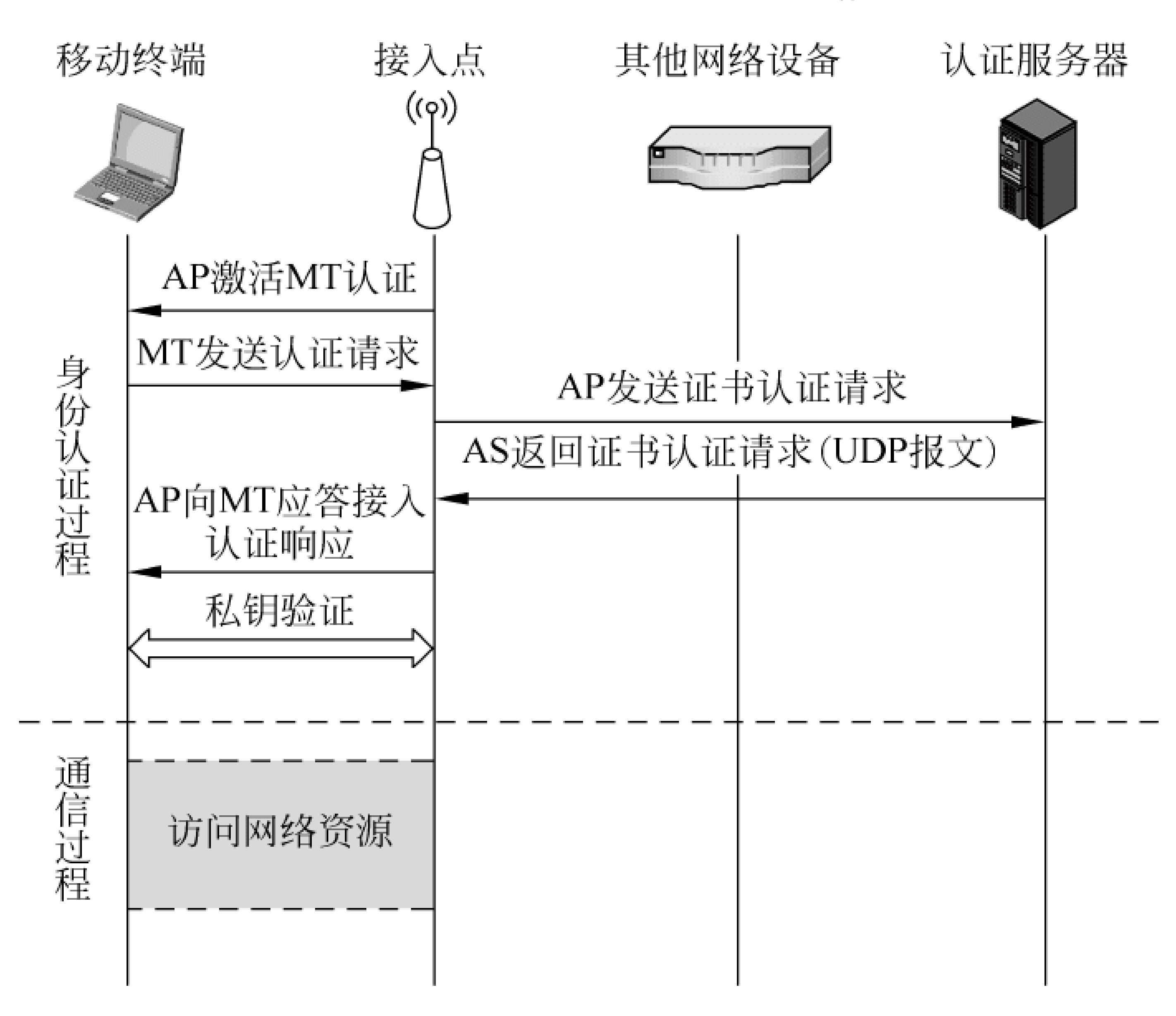


图 9.14 WAPI 的工作原理

图 9.14 显示的是 WAPI 的完整交互过程，下面结合该图来分析 WAPI 的工作流程。

① 认证发起过程。认证开始之前，移动终端首先要登录到 AP，建立链路连接。移动终端通过无线信道的链路连接建立成功接入到 AP 后，AP 向移动终端发送认证激活信息，以启动整个认证过程。

② 移动终端发送接入认证请求。移动终端向 AP 发出接入认证请求信息，接入认证请求信息的内容包括移动终端的证书、移动终端的当前系统时间。其中系统时间称为接入认证请求时间。

③ 接入点发送证书认证请求。AP 收到移动终端接入认证请求后，向认证服务器发出证书认证请求信息。证书认证请求信息的内容包括移动终端的证书、接入认证请求时间和 AP 证书，并利用 AP 的私钥对它们签名生成证书认证请求报文发送给认证服务器。

④ 认证服务器发送证书认证响应。当认证服务器收到 AP 的证书认证请求后，它将验证 AP 的签名及 AP 和移动终端证书的合法性。验证完毕后，认证服务器将移动终端证书认证结果信息(包括移动终端证书、认证结果及认证服务器对它们的签名)、AP 证书认证结果信息(包括 AP 证书、认证结果、接入认证请求时间及认证服务器对它们的签名)构成证书认证响应报文发回给 AP。

⑤ 接入点发送接入认证响应。AP 收到认证服务器的证书认证结果之后，AP 对认证服务器返回的证书认证响应进行签名验证，得到移动终端证书的认证结果。AP 将移动终端证书认证结果信息、AP 证书认证结果信息及 AP 对它们的签名组成接入认证响应报文回送至移动终端。移动终端验证认证服务器的签名后，得到 AP 证书的认证结果。移动终端根据该认证结果决定是否接入该 AP。

⑥ 私钥验证请求。这是一次双向的认证，AP 和移动终端都需要确认对方是否是证书的合法持有者。私钥验证请求包含实时产生的随机数，请求对方对其签名，以验证对方是否拥有该证书的私钥。该请求可由 AP 或移动终端发起。

⑦ 私钥验证响应。包含对私钥验证请求中随机数据的签名，提供自己是证书合法持有者的证明。

⑧ 至此移动终端与 AP 之间完成了证书认证过程。若认证成功，则 AP 允许移动终端接入，否则解除其登录。

在证书双向认证结束后，若 AP 和移动终端可以利用合法证书的公钥进行会话密钥的协商，上述的私钥验证过程也可省略，实现密钥的集中、安全管理。

9.4.2 WAPI 的特点

通过以上分析，可以看出 WAPI 的特点如下。

① 采用基于公钥密码体系的证书机制，实现了移动终端与无线接入点间的双向认证。

② 用户只需安装一张证书就可在覆盖 WLAN 的不同地区漫游，方便用户使用。AP 设

置好证书后，无须再对后台的 AAA 服务器进行设置，安装、组网便捷，易于扩展。

WAPI 具有几个重要特点：全新的高可靠性安全认证与保密体制，更可靠的二层（链路层）以下安全系统，完整的“用户-接入点”双向认证，集中式或分布集中式认证管理，灵活多样的证书管理与分发体制，可控的会话协商动态密钥，高强度的加密算法，可扩展或升级的全嵌入式认证与算法模块，支持带安全的越区切换；支持 SNMP 网络管理，完全符合国家标准，通过国家商用密码管理部门安全审查，符合“国家商用密码管理条例”。

由于会话密钥并没有在信道上进行传输，因此就增强了其安全性。为了进一步提高通信的保密性，WAPI 还规定，在通信一段时间或者交换一定数量的数据之后，STA 和 AP 之间可以重新协商会话密钥。WAPI 采用对称密码算法实现对 MAC 层 MSDU 进行的加、解密操作。

WAPI 充分考虑了市场应用，从应用模式上可分为单点式和集中式两种。单点式主要用于家庭和小型公司的小范围应用；集中式主要用于热点地区和大型企业，可以和运营商的管理系统结合起来，共同搭建安全的无线应用平台。因此，采用 WAPI 可以彻底扭转目前 WLAN 多种安全机制并存且互不兼容的现状，从而在根本上解决安全问题和兼容性问题。

9.5 移动 IP 安全概述

9.5.1 移动 IP 概述

由于 IP 路由基于网络前缀，IP 数据分组首先到达对应 IP 地址网络前缀的网段，然后转发到目的主机。因此，当主机在不同链路间移动时，如果其 IP 地址保持不变，那么它的物理位置就不能再由该 IP 地址来确定，所有发送给该主机的 IP 分组不能被正确地转发。在移动 IP 产生之前，曾有人提出几种方案来解决主机移动过程中的路由和通信的连续性问题。

① 在主机改变网络接入点的同时改变其 IP 地址。这种方法只有当主机主动发起通信时才能被网络中其他主机识别。但是不能保持主机现有的网络连接和通信的连续性，因为 IP 地址的改变导致所有正在进行的传输层连接都被中断。

② 根据特定主机 IP 地址进行路由选择。这种方法将占用路由器大量的资源，对于每个数据分组的选路，路由器都要进行大量的主机地址入口的搜索，使得系统可扩展性差，不能满足大规模网络互联的要求。

③ 在数据链路层解决移动问题。蜂窝数字分组数据网（cellular digital packet data，CDPD）的用户申请服务时，由电信 ISP 为其分配一个在整个 CDPD 网络中使用的 IP 地址，由数据链路层保证分组能按照 ISP 提供的 IP 地址准确送达。虽然其概念上与移动 IP 相

似，但是它只能在 CDPD 系统中提供移动性，无法与现有的 Internet 兼容。与 CDPD 相比，IEEE 802.11 是一个在地域范围上受限制但速率更高的链路层解决方案。在 IEEE 802.11 中，设备的移动对 IP 层是不可见的。然而，如果节点移动时穿过了一台路由器，那么它就应该改变 IP 地址，这样正在进行的通信就会中断。因此，单纯依靠数据链路层解决方案是不可行的。

IETF 的移动 IP 工作组于 1992 年初步制定了工作在网络层的移动 IP 协议，并由 IESG（Internet engineering steering group）在 1996 年 6 月通过移动 IP 的标准草案，同年 11 月公布为建议标准，为移动 IP 成为 Internet 正式标准奠定了基础，对移动 IP 的发展起到了关键性的作用。由于 IPv4 的广泛应用，目前只有移动 Ipv4 标准协议以 RFC 形式发布。而移动 IPv6 仍然处于不断的修订之中。由于 IPv4 与 IPv6 之间的差异，使得移动 IPv4 协议与移动 IPv6 协议有着本质上的区别。2003 年，IETF 的移动 IP 工作组分为两个独立的工作组，分别负责制定移动 IPv4 标准和移动 IPv6 标准。

移动 IP 协议是目前公认支持通信节点移动的网络层解决方案，它与数据链路层协议和物理传输介质无关。下面简述移动 IP 的基本原理。

移动 IP 的主要设计目标是移动节点在改变网络接入点时，不改变其 IP 地址而保持其通信的连续性。具体地说，移动 IP 协议的设计应当满足如下要求。

① 移动节点在改变数据链路层的接入点之后，能够保持与 Internet 上通信节点通信的连续性。

② 无论移动节点连接何种数据链路层接入点，它都能够使用原来的 IP 地址进行通信。

③ 移动节点能够与不具备移动 IP 功能的通信节点进行通信。

④ 移动节点不应该比 Internet 上的其他节点面临新的或更多的安全威胁。

此外，由于移动节点通常通过无线链路连接到 Internet 上，而无线链路具有低带宽、高误码率的特点，而且作为移动节点的移动计算机电池能量有限，如何减少通信过程中的能量消耗也是需要重点考虑的问题。因此，设计移动 IP 协议时，要考虑移动节点接入网络时发送的管理消息数目应尽可能少，同时采用尽可能短的消息进行通信。

9.5.2 移动 IP 的工作原理

下面首先介绍移动 IP 协议中重要的实体。

- 移动节点（mobile node，MN）：是需要进行移动并可以使用原有 IP 地址（家乡代理的永久地址）和其他通信节点进行通信的节点。它需要支持移动 IP。
- 家乡代理（home agent，HA）：是指有一个端口与移动节点的家乡链路相连的路由器。移动节点以其转交地址（care of address）通知家乡代理它的当前位置。家乡代理广播对移动节点的家乡地址网络前缀的可达性，可获取那些发往移动节点的家乡地址的 IP 分组，并将这些 IP 分组通过隧道技术传送到移动节点的转交地址。

- 外地代理(foreign agent,FA):是指在移动节点所移动至的外地链路上的路由器。外地代理帮助移动节点将其转交地址通知家乡代理。某些情况下,外地代理提供移动节点的转交地址,并为已被家乡代理设置隧道的移动节点发送拆封后的 IP 分组。对于移动节点发出的 IP 分组,外地代理提供类似于默认路由器的服务。

某些情况下,将家乡代理和外地代理统称为移动代理。图 9.15 给出了移动 IP 各功能实体之间的相互关系。

图 9.15 移动 IP 各功能实体之间的相互关系

移动 IP 的其他重要术语如下。

- 家乡网络(home network):是指与移动节点的家乡地址具有相同网络前缀的网络,也可以是一个物理上不存在的虚拟网络。发往移动节点的家乡地址的 IP 分组以标准的 IP 路由机制转发到其家乡网络上。
- 家乡链路(home link):是指与移动节点的家乡地址具有相同网络前缀的链路,是移动节点在家乡网络时的链路。家乡链路比家乡网络更为精确地描述了移动节点在家乡的位置。
- 外地网络(foreign network 或 visited network):是指除移动节点的家乡网络外的任何网络,也就是网络前缀与移动节点的家乡地址网络前缀不同的网络。
- 外地链路(foreign link):是指除家乡链路以外的链路,也就是网络前缀与移动节点家乡地址网络前缀不同的链路。外地链路比外地网络更为精确地描述了移动节点移动时的位置。
- 家乡地址(home address):是指每个移动节点在家乡链路上拥有的一个"长期有效"的 IP 地址。对这种地址的管理类似对固定主机 IP 地址的管理。
- 转交地址(care of address):是指当移动节点离开家乡网络后,被赋予的反映其当前链路接入点的临时 IP 地址。
- 通信对端节点(correspondent node):是指与移动节点通信的对等实体,以后简称为通信对端。通信对端可以是移动节点或者固定节点。
- 移动绑定(mobility binding):是指由家乡代理维护的移动节点的家乡地址和转交地址的关联,还包括关于该关联的剩余生存期等其他信息。

移动 IP 协议中，移动节点在移动过程中经历代理发现、注册、分组路由和注销等几个过程。

1. 代理发现(agent discovery)

家乡代理和外地代理周期性地在它们作为移动代理的链路上，多播或广播代理通告(agent advertisement)消息，通告它们与相应链路的连接关系。代理通告是通过在 ICMP 路由器通告消息中增加了"移动代理通告扩展"部分，来说明移动代理是家乡代理还是外地代理，以及它的网络地址和通告有效期等信息。

移动节点根据收到的代理通告消息，判断它是在家乡链路上还是在外地链路上。当连接在家乡链路上时，移动节点就和固定节点一样工作，不再启用移动 IP 的其他功能。当移动节点发现它从家乡链路移动到外地链路，或者从一个外地链路移动到另一个外地链路上时，它就要向家乡代理进行注册。

2. 注册(registration)

当移动节点连接在外地链路上时，它需要一个代表它当前所在位置的转交地址。移动节点可以从外地代理通告消息中获得外地代理的转交地址，或通过动态主机配置协议 DHCP 和手工配置等方法配置转交地址。

移动主机在获得转交地址后，通过移动 IP 协议定义的注册请求(registration request)消息向家乡代理注册。家乡代理确认后，将家乡地址和相应的转交地址存放在绑定缓存中，完成移动节点的家乡地址和转交地址的绑定，并向移动节点发送注册应答(registration reply)消息。在注册过程中，如果移动节点使用外地代理的转交地址，就要通过外地代理进行注册请求和注册应答。

图 9.16 给出了一个移动节点的注册过程示例。图中移动节点的家乡地址为 128.119.40.186，家乡代理 HA 的地址为 128.119.40.7，外地网络(visited network)为 79.129.13/24，外地代理给移动节点分配的转交地址 COA 为 79.129.13.2。

初始时，外地代理向移动节点发出一个通告消息(图 9.16 中的 ICMP agent advertisement)，其中包含分配给该移动节点的转交地址 COA。移动节点获得该 COA 地址后，向外地代理发出注册请求(图 9.16 中的 registration request 消息)，其中包含移动节点的家乡代理及其本地地址，以及生命周期等其他相关信息。该注册消息将被外地代理转发给移动节点的家乡代理，家乡代理完成移动节点的地址绑定和注册后，向外地代理发送注册应答消息(图 9.16 中的 registration reply 消息)，最后该应答消息被转发给移动节点，注册过程结束。

3. 分组路由(packet routing)

家乡代理和家乡链路上的其他路由器通过与外地链路上的路由器交换路由信息，使得

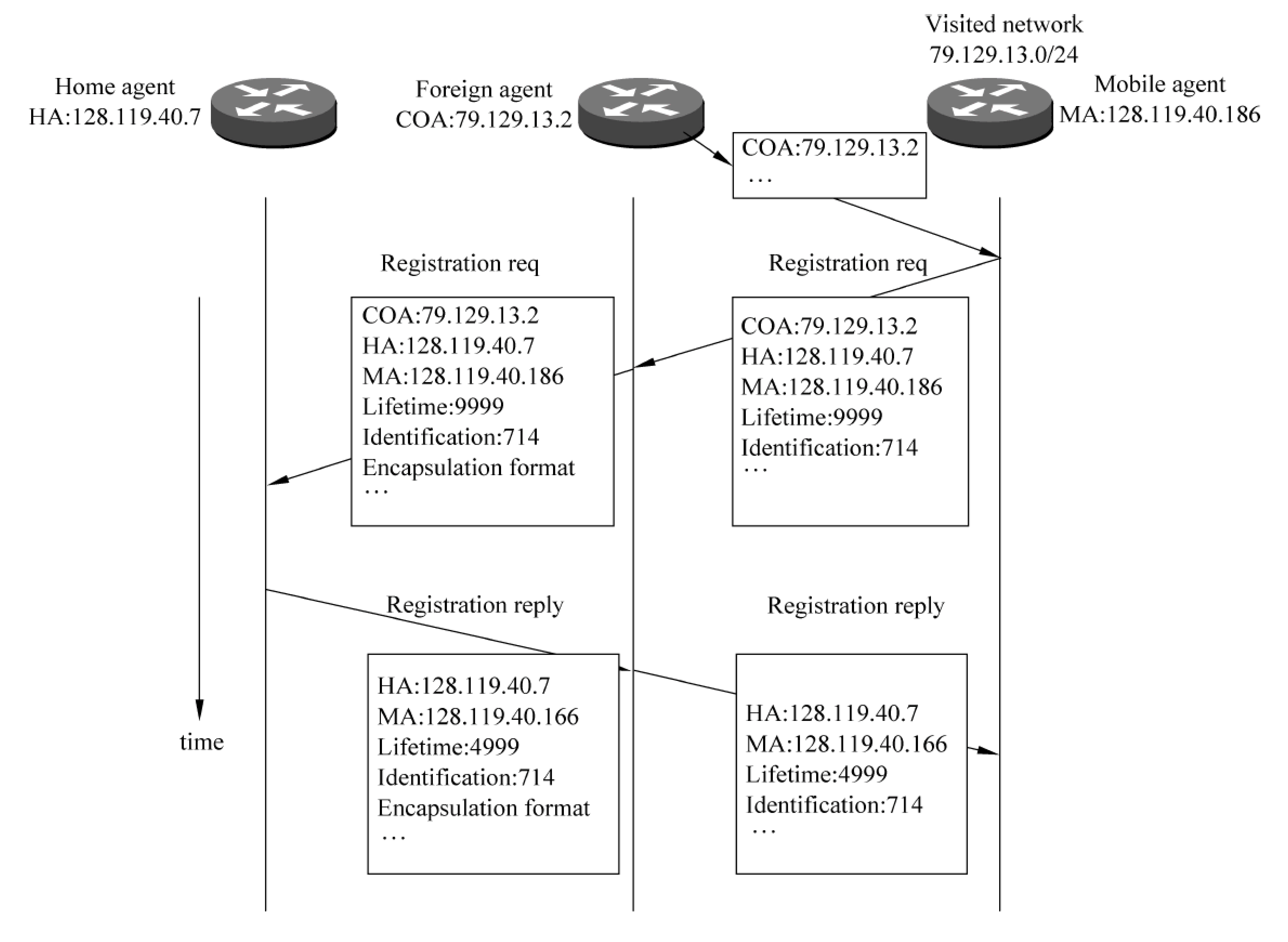


图 9.16 移动节点注册过程示意图

发送给移动节点的家乡地址的 IP 分组被正确地转发到家乡链路上。家乡代理通过地址解析协议来截取发往移动节点的家乡地址的 IP 分组。

家乡代理根据 IP 分组的目的 IP 地址查找绑定缓存，获得移动节点注册的转交地址，然后通过隧道发送该分组到移动节点的转交地址。如果转交地址是外地代理的转交地址，隧道末端的外地代理拆封得到原始分组后，转发给移动节点。如果转交地址是一个普通的"配置转交地址"，封装后的数据分组直接发送至移动节点。

移动节点使用外地网络的路由器作为默认路由器，它发送的 IP 分组通过外地网络的路由器直接发送给通信对端，无须采用隧道机制。这样，通信对端发送的 IP 分组通过移动节点的家乡代理转发给移动节点，移动节点的 IP 分组直接发送到通信对端，形成三角路由现象。三角路由不是优化路由，路由优化问题也是移动 IP 的一个比较重要的问题，主要工作是如何使通信对端在大多数情况下把 IP 分组直接发送到移动节点的转交地址，而不是总通过家乡代理来转发，从而保证 IP 分组的路由是优化的。

4. 注销(deregistering)

移动节点收到代理通告消息,如果判断它返回到家乡链路上,那么移动节点必须直接注册到家乡代理来注销它当前的转交地址。注销之后,移动节点就可以像固定节点一样工作,直到它再次离开家乡链路。

9.5.3 移动 IP 面临的安全威胁及对策

移动 IP 面临的安全问题源于多种因素。首先,移动 IP 是工作在网络层的协议,它所引入的新的控制消息,如代理通告、注册请求和应答、绑定更新和代理发现等如果处理不当,容易受到攻击。此外,移动 IP 虽然可以工作在任何种类的链路上,但是在大多数情况下,移动 IP 应用于无线网络环境中,也就是通过无线链路接入网络,无线网络的特殊性使得同样的攻击在无线网络中更容易实施。再者,移动 IP 中的移动节点常常离开家乡网络,在不同的外地网络之间漫游,由于无法保证所有的外地网络都是可信的,因此移动节点很容易遭受被动窃听、会话窃取和各种主动攻击。

针对移动 IP 协议自身的特点及移动环境的特殊性,移动 IP 可能遭受各种安全攻击,包括拒绝服务攻击、重放攻击、中间人攻击、会话窃取和被动窃听等。下面简要介绍这些攻击行为及相应的移动 IP 安全机制。

1. 拒绝服务攻击

在移动 IP 协议中,注册的一个首要目的是让移动节点将其转交地址通知其家乡代理,家乡代理将接收那些目的地址为移动节点家乡地址的 IP 分组,并通过隧道发送到移动节点的转交地址。攻击者可以发出一个伪造的注册请求消息给家乡代理,以其 IP 地址或一个欺骗 IP 地址代替移动节点的转交地址。这样,通信对端发出的所有 IP 分组都会被移动节点的家乡代理通过隧道发送至攻击者指定的 IP 地址,而不是移动节点的转交地址。如图 9.17 所示,移动节点不再能够接收任何 IP 分组,它当前的所有通信都将被中断,导致拒绝服务。

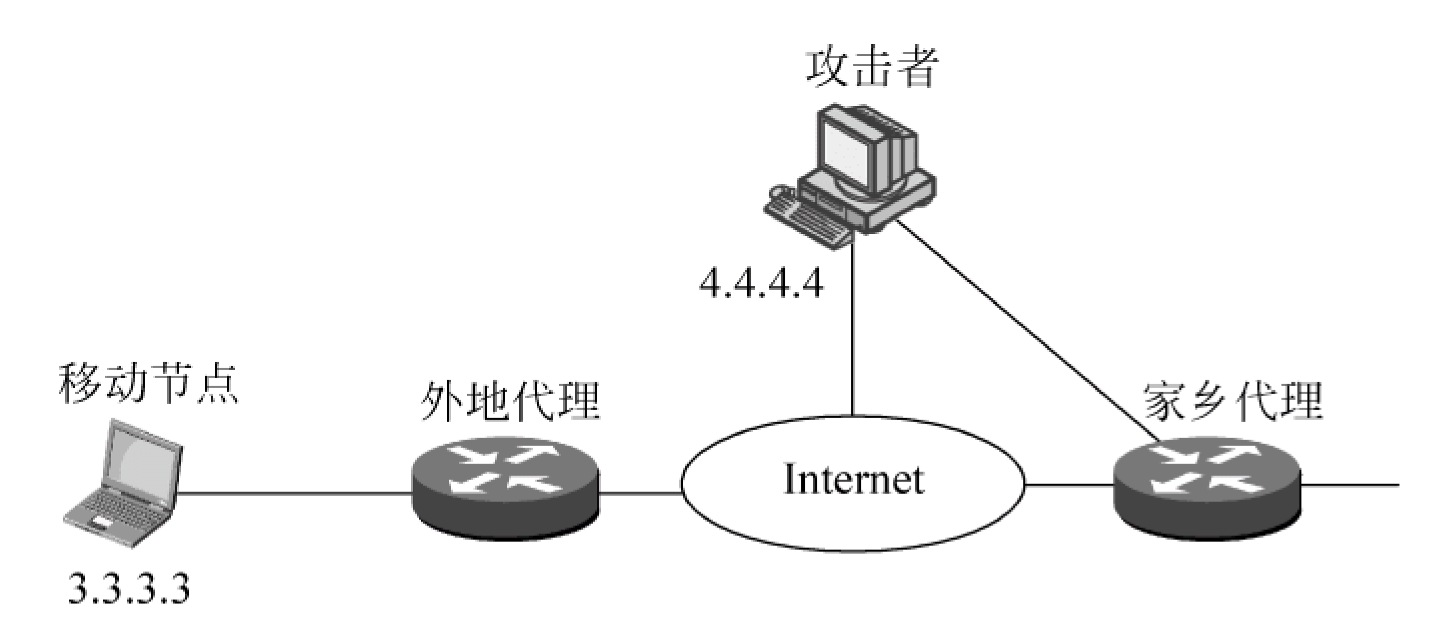


图 9.17 移动 IP 中的拒绝服务攻击

同样,攻击者可以通过假冒外地代理对移动节点发起拒绝服务攻击。为了防御这种移动 IP 的拒绝服务攻击,可以对移动节点和家乡代理之间的注册消息进行认证。为此,移动 IP 协议提供了一些认证机制:通过认证扩展的方式提供了移动节点和移动代理之间的注册消息的认证,即移动—家乡认证扩展,移动—外地认证扩展,外地—家乡认证扩展。其中,移动—家乡认证扩展是必选的,其余两个是可选的。图 9.18 给出了包含认证扩展的注册消息的一般结构。

IP 报头(RFC 791)
UDP 报头(RFC 768)
注册消息的移动 IP 字段(RFC 3344)(定长部分)
可选扩展……
移动—家乡认证扩展(RFC 3344)
更多的可选扩展……

图 9.18 包含认证扩展的注册消息

移动 IP 协议使用的默认认证算法是 HMAC—MD5,采用前缀加后缀的模式。以移动—家乡认证为例,图 9.19 给出了利用 HMAC—MD5 注册消息的认证过程。

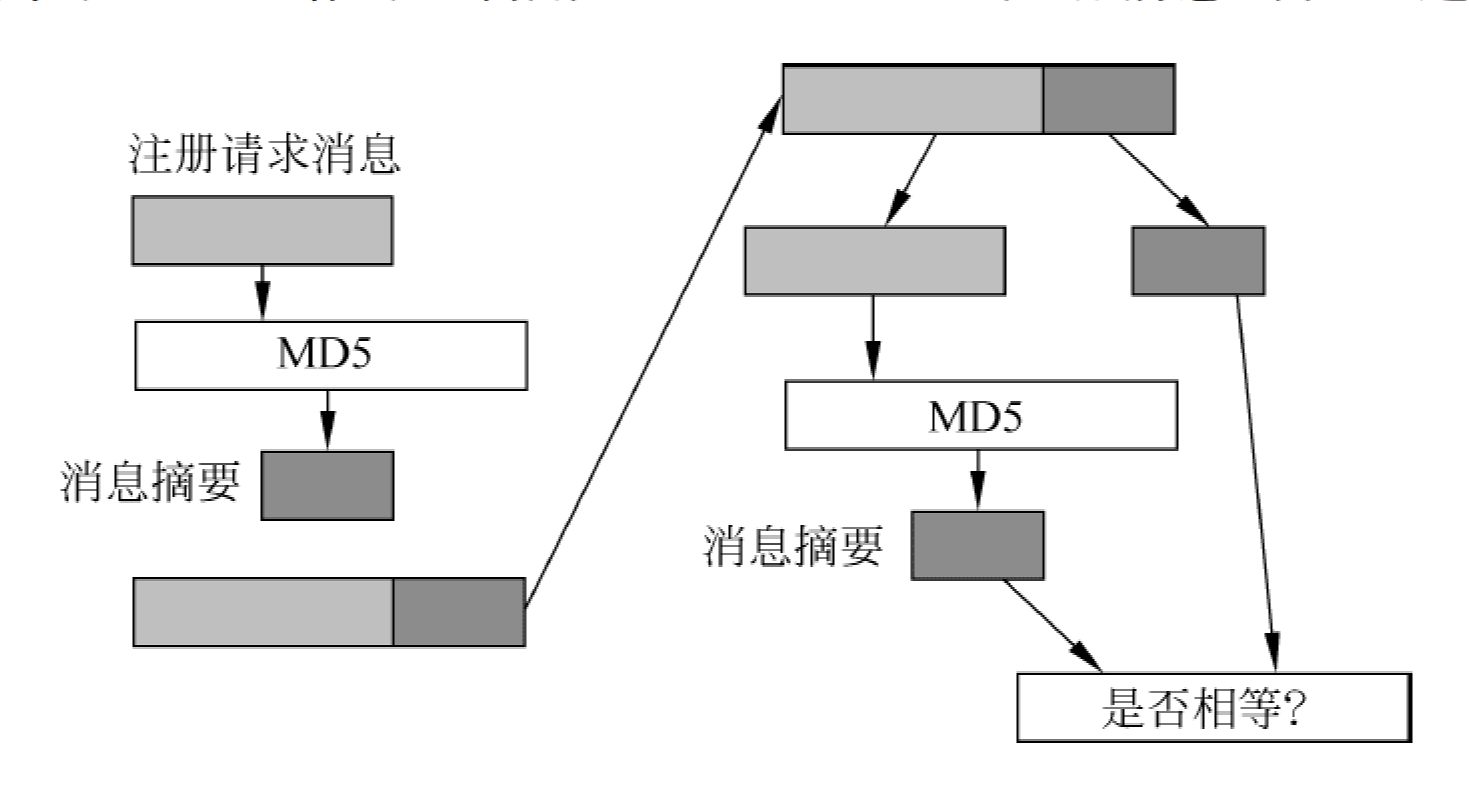


图 9.19 采用 HMAC-MD5 的注册消息认证过程

移动节点产生一条注册请求信息,其中包括固定部分和移动—家乡认证扩展,移动节点填写请求消息和扩展部分中除认证域外的所有其他字段,然后对以下这一字节串计算出一个消息摘要:共享密钥,注册请求消息的定长部分,除认证域外包括移动—家乡代理认证扩展在内的所有扩展(即类型、长度和安全参数索引),接着是移动节点和家乡代理的共享密钥。移动节点将这个消息摘要写入移动—家乡认证扩展的认证字段中,这样就完成了一个注册请求消息的封装。然后,移动节点将这个消息发送给家乡代理。

当注册请求消息到达家乡代理后,家乡代理所做的工作与移动节点在组装消息时所做的工作大致相同。家乡代理利用它和移动节点共享的秘密密钥及接收到的注册请求消息的各个字段计算消息摘要,然后将计算结果与从移动节点那里接收到的认证字段进行比较。

若相等则注册认证成功,移动 IP 的认证扩展同时提供了认证和完整性检验。家乡代理向移动节点返回注册应答时的过程正好相反。家乡代理计算注册应答消息和密钥的消息摘要,将消息摘要放在注册应答的认证字段中,移动节点检查消息摘要来对家乡代理进行认证,并检查消息的完整性。

2. 中间人攻击

中间人攻击是指攻击者拦截网络中的分组,经过修改之后再送回到网络中。在移动 IP 的代理发现机制中,移动节点可能会遭受中间人攻击。

在移动 IP 中,移动代理周期性地发送代理通告消息,移动节点根据收到的代理通告消息来判断自己的位置,判断是在家乡链路还是在外地链路。代理通告消息是作为 ICMP 路由器发现消息的移动扩展发送的。其中包含一个序列号,移动节点根据该序列号判断外地代理是否重启,也就是说外地代理不再知道该移动节点的存在,需要重新注册。攻击者可以利用这种机制,伪造一个代理通告消息,使得收到这个消息的移动节点认为需要重新注册。一般代理通告消息中的 TTL 字段的值为 1 时,移动节点才接受该消息。所以攻击者通常与移动节点位于同一链路上,但是它也可以在其他链路上,通过适当的设置 TTL 字段的值来使移动节点收到该消息时,使其中的 TTL 字段为 1。

代理通告消息的发送方式可以是广播、多播和单播。由于广播和多播只能在当前链路上发送和接收,因此,当攻击者位于移动节点所在链路以外的其他链路上时,这种攻击是无效的。而当移动节点发送代理请求消息时,由移动代理响应的代理通告是以单播形式发送给移动节点的。这时,如果攻击者伪造一个单播的代理通告地址给移动节点,就有可能成功。因此,移动节点应该抛弃那些单播的代理通告消息,除非它发送了代理请求消息。

3. 重放攻击

重放攻击是指攻击者截获数据,等待一段时间后重新发送,一般表现为对认证系统的攻击。攻击者记录下任何保持不变的认证信息,比如密码,然后重放以前发送过的合法消息,以此来骗过认证系统。

在移动 IP 的注册过程中存在遭受重发攻击的隐患。一个攻击者可以将一个有效的注册请求信息保存起来,然后经过一段时间后再重放这个消息,从而注册一个伪造的转交地址。为防止这种重放攻击,移动节点为每一个连续的注册消息标识(identification)字段产生一个唯一值。该值使得家乡代理可以知道下一个值应该是多少。这样,攻击者就无能为力了,因为它保存的注册请求信息会被家乡代理判定为已经过时了。

移动 IP 定义了两种填写标识字段的方法:时间戳(必须的)和 nonce(可选的)。在移动节点与家乡代理之间有效的重放保护的形式是移动安全关联的一部分。一个移动节点与家乡代理必须就将采用哪种重放保护方式达成一致。对于标识字段的解释取决于重放保护的方法。无论采用哪种方法,标识字段的最低 32 位必须从注册请求消息中原封不动地复制到

注册应答消息中。外地代理使用这些位(和移动节点的本地地址)来比较注册请求与相应的注册应答。移动节点必须验证注册应答消息的标识字段的低 32 位与注册请求消息中的该字段是否相同。

(1) 基于时间戳的重放保护

时间戳重放保护的基本原理是：节点 A 产生一个消息插入当前的日期和时间，收到消息的节点 B 检查这个时间戳是否和它自己的日期时间足够接近。除非在节点之间的安全关联中详细说明，将使用默认值 7s 作为时间差别的限度，而且应该至少大于 3s。很显然，两个节点必须拥有很好的经过同步的时钟。与其他信息一样，时间同步信息也可以根据两个节点间的安全关联而采用某种认证机制来防止被篡改。

如果采用了时间戳协议，移动节点必须将标识字段设置为由网络时间协议(network time protocol，NTP)所指定的一个 64 位的数值。NTP 格式的低 32 位代表秒的小数部分，这些位不能从一个时间源得到，而应该由一个好的随机源产生。

需要注意的是，当使用时间戳时，在一个注册请求消息中所使用的 64 位标识字段的值必须大于任何先前的注册请求信息中的标识字段的值，因为家乡代理同时要使用这个字段作为一个序列号。如果没有这样的序列号，很可能会将一个早些时候的注册请求延迟了的副本送给家乡代理(在家乡代理所要求的时钟同步限制之内)，造成次序颠倒，从而错误地改变了移动节点当前注册的转交地址。

一旦收到了带有可授权扩展的注册请求后，家乡节点必须检验标识字段的合法性。标识字段内包含的时间戳必须足够接近家乡代理的时钟，并且时间戳必须大于所有先前接受过的目前正在请求的移动节点的时间戳。时间公差和重同步的详细情况取决于某个特定的移动安全关联。

如果时间戳是合法的，家乡代理把整个标识字段复制到返回给移动节点的注册应答消息中去。如果时间戳不合法，家乡代理仅仅把低 32 位复制到注册应答消息中，并用高 32 位提供它自己的时钟。在后一种情况下，家乡代理必须通过在注册回复中返回 Codel 33 来拒绝注册。

移动节点在使用注册应答消息中的标识字段的高 32 位进行重同步之前，必须证实在注册应答消息中标识字段的低 32 位与那些被拒绝的注册请求中的相同。

(2) 基于 nonce 的重放保护

nonce 重放保护的基本原理是：节点 A 在发给节点 B 的每一个消息中包含一个新的随机数，并检查在节点 B 发给 A 的下一个消息中返回的数值是否相同。这两个消息都使用一个认证码来防止某个攻击者对其篡改。同时，节点 B 可以在所有发送给节点 A 的消息(由节点 A 来响应)中包含它自己的 nonce，因此它也能证实所收到消息的新鲜性。

家乡代理在每一个注册应答中插入一个新的 nonce 作为标识字段的高 32 位。家乡代理把注册请求消息中标识字段的低 32 位复制到注册应答消息中标识字段的低 32 位中。当移动节点从家乡代理那里收到一个经过认证的注册应答消息时，它把标识字段的高 32 位保

存下来，作为下一个注册请求信息的高 32 位。

移动节点负责在每一个注册请求中产生标识字段的低 32 位。理想情况下，它应该产生自己的随机数 nonce。但是它可以使用其他合适的方法，包括复制家乡代理发送的随机数值。所选择的标识字段的高 32 位和低 32 位都应该与它们先前的值有所不同。在每次的注册消息中，家乡代理使用一个新的高位值而移动节点使用一个新的低位值。外地代理使用低位值（以及移动主机的家乡地址）来正确匹配注册应答和未决的请求。如果一个注册请求消息因为一个非法的 nonce 而被拒绝，注册应答消息总是为移动节点提供一个新的 nonce 用作下一次的注册。

4. 会话窃取与被动窃听

会话窃取攻击是指攻击者在一个合法节点进行认证并开始会话后，通过假冒合法节点将会话窃取过去。通常，攻击者必须发送大量的无用数据来防止合法节点发现会话已被窃取了。

移动 IP 的注册过程可能遭到攻击者的会话窃取攻击。假设攻击者与移动节点位于同一链路上，他首先等待移动节点向家乡代理进行注册。然后攻击者偷听到移动节点开始了一个他所感兴趣的通信会话，便发送大量无用的数据分组给移动节点，占用移动节点的全部 CPU 资源。攻击者发送假冒移动节点发出的消息给移动节点的通信对端，并截获发往移动节点的数据分组，从而成功地窃取会话。这时，移动节点当前的通信被中断了，但是并不知道它当前的会话已经被攻击者窃取过去。

为了防止这种攻击，要求移动节点和外地代理之间存在链路层加密，最好是端到端的加密。在外地链路是无线的情况下，数据链路层加密尤为重要。在无线链路上进行会话窃取要比在有线环境下容易，因为攻击者不需要物理地连接到链路上。进行端到端的加密是更好的选择。端到端加密是指在通信源端对数据进行加密，在目的端对数据进行解密，而不只是在某一段链路上对数据进行加解密。这样的优点在于，在网络中任何一点数据都是加密的，而不只是在外地链路上得到保护。数据的加密与物理介质无关，而且加解密只在通信端点进行，而不是在通信路径中的某些地方，这样就防止了不必要的时延。

被动窃听是指攻击者窃听并截取网络中传输的数据分组，以窃取数据分组中可能包含的机密和私人信息。防范这种攻击最好的方法就是对数据进行加密。这和前面对付会话窃取攻击的方法一样。

移动 IP 中的移动安全关联（mobile security association）是一组用于保护消息的安全策略。两个移动实体进行安全通信前，必须首先协商一个安全关联，选择通信双方都能支持的加密与认证算法。移动安全关联由以下几部分组成：加密算法（如 DES、3DES、Blowfish、CAST 和 AES 等）；消息摘要算法（如 MDS、SHA 和 Tiger 等）；认证算法（如预先分配共享密钥，数字签名和共享密钥等）；移动安全关联的生存期。

9.6 移动 IP 安全机制

在了解移动 IP 面临的安全威胁之后，在设计移动 IP 的安全体系时，应考虑移动 IP 协议特殊应用环境中的如下问题。

① 可扩展性。在移动 IP 协议的应用环境中，移动节点在不同网络之间漫游，其通信对等节点也分散在不同的网络中，这些网络采取的安全机制可能各不相同，而且网络之间不存在必要的安全信任关系。这就要求移动 IP 的安全机制具有较好的可扩展性，以满足各种不同的网络安全机制。

② 兼容性。在移动 IP 协议中，通信对端可以是网络中的任何节点，这些节点可能具有固定网络中的安全机制，移动 IP 协议的安全体系不应该影响其原有的安全机制。

③ 复杂性。使用移动 IP 协议的移动设备的计算能力和能量较低，因而移动 IP 的安全体系应该尽可能简单，计算量小。安全体系还要考虑移动节点快速移动中通信量的切换问题。为了减少切换造成的延迟，要求安全协议进行消息交换的次数要少，最好能够与移动 IP 协议的消息一起传输。

根据移动 IP 协议自身的特点，以及特殊的应用环境，有人提出了一些保护移动 IP 协议的安全性的方案，以弥补移动 IP 协议安全能力的不足，其中包括采用 AAA 框架的认证系统、采用公钥体制的方案，以及解决移动 IP 穿越防火墙的方案等。下面介绍几种具有代表性的移动 IP 的安全技术。

9.6.1 基于 AAA 的移动 IP 认证机制

可以使用 AAA 框架实现对移动 IP 的认证，利用公钥体制实现认证协议。该协议具有如下假定。

① 在移动节点与外地 AAA 服务器 AAAF(foreign AAA server)之间没有安全关联。

② 移动节点与家乡 AAA 服务器 AAAH(home AAA server)共享一个基于秘密密钥的安全关联。

下面给出该协议的注册过程。

① 外地代理发现。移动节点一旦进入某个外地网络，就必须与该网络中的外地代理取得联系。为此，移动节点必须能够识别外地代理。外地代理以一定时间间隔发送代理通告消息，其中包括外地代理认证协议列表和 AAAF 标识扩展。

② 移动节点注册请求的产生。移动节点一旦收到外地代理通告消息，便产生一个注册请求。除了注册请求的定长部分，还包含移动节点的网络访问标识符(network access identifier, NAI)，移动节点 API 扩展，并将 API 字段设置为 3 表示 Min PKA 协议。同时还有 AAA

标识扩展，封装的外地代理通告(encapsulated foreign agent advertisement，EFAA)，以及用移动节点与AAAH共享的秘密密钥产生的消息认证码。

③ 外地代理对注册请求的处理。外地代理收到由移动节点用外地代理的公钥加密的注册请求之后，用它自己的私钥解密该消息。外地代理不需要进行认证，它只是观察EFAA来确保该注册请求是对最近发出的代理通告的响应。一旦成功验证EFAA，外地代理就将移动节点的注册请求转发给AAAF。

④ AAAF对注册请求的处理。收到来自移动节点的注册请求之后，AAAF在该请求消息之后附加外地——家乡认证信息。该认证信息包括由AAAF产生的一个对AAAH的询问(挑战字)。同时还包括AAAF证书的副本及对所有请求消息字段的一个数字签名。然后AAAF将该请求转发给AAAH。

⑤ AAAH对注册请求的验证。AAAH用AAAF的证书中的公钥解密该消息。这就可以证明该消息是由拥有一对公钥私钥的节点产生的，同时证书可以进一步确保该消息由AAAF产生。AAAH用它与移动节点共享的秘密密钥验证移动节点的注册请求消息中的消息认证码。这样，AAAH就对移动节点和AAAF的身份进行了认证。于是，AAAH将必要的注册请求信息转发给家乡代理来更新移动节点的移动绑定。

⑥ AAAH注册应答的产生。AAAH产生一个注册应答消息，其中包括移动节点的NAI，AAAH与移动节点共享的秘密密钥产生的消息认证码的移动——家乡AIM扩展，AAAF产生的挑战字及AAAH的证书，然后是对整个应答消息字段产生的数字签名。

⑦ AAAF对注册应答的认证。AAAF收到注册应答之后，用AAAH证书中的公钥解密该消息。验证AAAH响应了一个正确的挑战应答值。然后通过安全链路转发该消息给外地代理。

⑧ 外地代理对注册应答的处理。外地代理收到来自AAAF的注册应答之后，直接转发给移动节点。

⑨ 移动节点对注册应答的处理。移动节点收到来自外地代理的注册应答消息，验证移动——家乡认证扩展中的消息认证码的合法性。

基于移动IP协议中的信任关系，该协议假定AAAH的验证可以证明注册请求来自合法的移动节点。移动节点也依赖于AAAH的认可来保证AAAF身份的真实性及被访问网络能够提供合格的服务。对AAAH身份的合法性的验证是通过AAAH产生的注册应答消息的数字签名来实现的。AAAH能够通过消息认证码来确保注册请求消息来自一个合法的移动节点。同时，AAAH通过检查注册请求中AAAF的数字签名来对AAAF进行认证。而移动节点能够确信注册应答确实来自AAAH，因为在注册应答中包含合法的消息认证码。为了防止重放攻击，该协议要求在AAAF的注册请求和AAAH的注册应答中必须包含AAAF的挑战字。

该协议提供了移动实体之间的认证，保护了移动IP的注册消息。从该协议中可以看出，有关公钥的计算主要集中在计算能力相对较强的AAA服务器和移动代理，而移动节点

则几乎不需要进行公钥计算。但是由于引入了公钥体制，当应用于无线环境中时，仍然可能会产生额外的延迟。所以该协议在消息延迟方面不具优势。

9.6.2　基于公钥的移动 IP 安全构架

John Zao 等人提出了一种称为 MoIPS(mobile IP security)的公钥管理系统，可以用于 IETF 的移动 IP 标准协议。MoIPS 提供了对移动 IP 的管理消息的认证，如移动节点的位置更新，对外地网络资源的访问控制及重定向 IP 分组的安全隧道。为了提供这些服务，该方案的作者提出了一种基于公钥的系统结构。

在基本移动 IP 协议中的注册消息和路由优化的移动 IP 协议中的位置绑定消息中都包含位置绑定信息。一个攻击者可以通过改变这些位置绑定信息，创建伪造的消息，或者重放预先录下的消息，将消息流向从一个节点重定向到另一个节点。为了克服这些问题，必须为注册消息和绑定更新消息提供数据认证，数据源认证和防止重放攻击的服务。因此，该方案提议使用一个 64 位的标识字段来抵御重放攻击，以及一个或多个认证标识字段来提供消息完整性和消息源认证。

为了获得访问控制，移动节点必须完成注册并得到对被访问网络的一个接入点。为此必须证实移动节点的身份和当前的状态。在该协议中，通过交换公钥证书可以验证移动节点的身份。移动节点的状态可以通过在外地代理和家乡代理之间交换经过认证的注册请求和注册应答消息来隐含地得到。

当移动节点漫游到远离家乡网络的地方时，将需要安全隧道来传输来往于移动节点和家乡网络的数据。同样，外地代理要为移动节点转发数据，在外地网络中的数据也需要有一个经过认证并且可信的实体如家乡代理来重定向。这些安全隧道可以采用 IPSec 中的隧道模式来实现。

该方案中基于公钥的管理系统包含以下三个安全方面的支持。

① 一个可升级的密钥管理系统，能够为任意一对节点产生并分配长期的密钥参数。

② 一个快速的短期密钥产生算法，用于提供对移动 IP 注册和绑定更新消息的认证所需要的短期密钥。

③ 移动 IP 协议与 IPsec 协议的协作。

John Zao 等人开发一个公钥管理系统来管理那些签发给 Internet 节点的公钥证书和证书撤销列表。他们还选择使用 Internet 域名服务系统 DNS 作为主要的证书仓库。之所以选择 DNS，是因为 Internet 节点是通过域名或 IP 地址来标识的，而两者都由 DNS 来保存。因此，当通信是由 DNS 查找建立起来的时候，DNS 证书的获取可以通过正常的网络实体之间的消息交换来实现。

使用公钥基础设施 PKI 技术的主要优点是可扩展性。基于 DNS 的 PKI 系统与密钥分发中心(key distribution center，KDC)相比有着明显的优势，这是因为 DNS 解决了潜在的

复杂的服务器发现问题，而且经过鉴定的长期公钥排除了对实时密钥分发的需求。公钥是离线地由证书机构签发，而 KDC 是在线地引入了可信任的第三方，这使得可扩展性成为系统的瓶颈。图 9.20 给出了 MoIPS 的系统结构图。

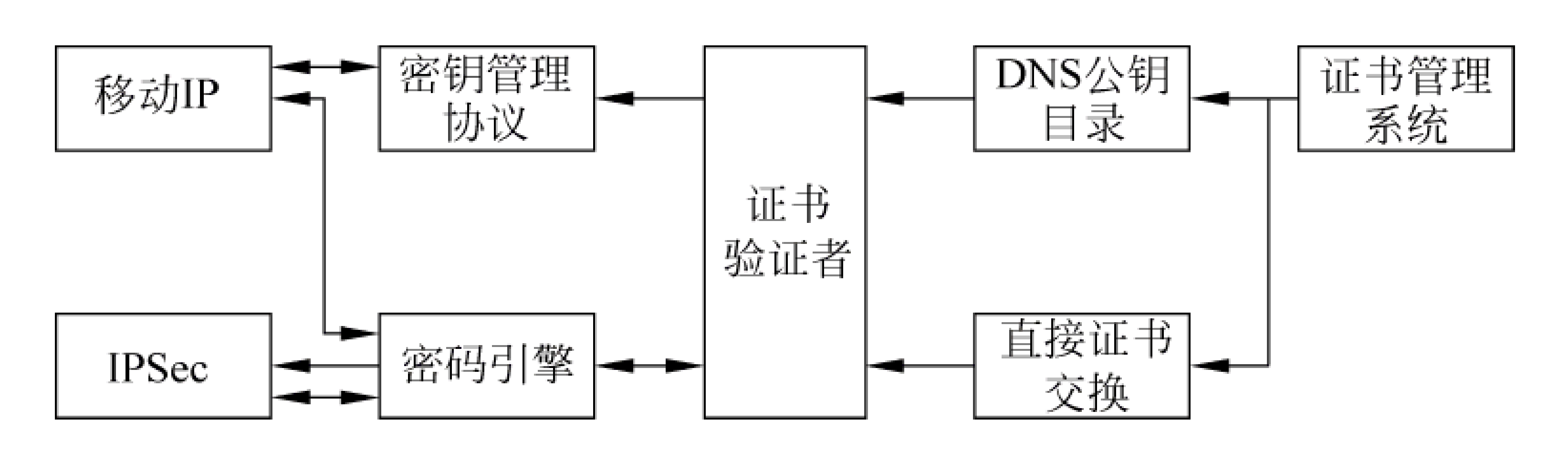


图 9.20 MoIPS 系统结构框图

在 MoIPS 系统中，移动 IP 和 IPSec 模块都利用了一个密钥管理模块和一个密码引擎。密钥管理模块产生安全服务所需的长期密钥，而密码引擎则执行实际的密码操作。密钥和其他安全参数保存在一个受保护的数据库中传送给密码引擎。证书验证者维护一个接受的和已验证的证书缓存，来减少证书获取和签名验证操作的数量。为了便于集成不同开发商的各种证书管理系统，MoIPS 没有包括 CA 的实现。

X.509 证书中有一些比较重要的字段，如 IssuerALtName 用于证书等级的建立，CertPolicy 允许包含特殊策略信息来指出一个节点是移动节点还是固定节点，是家乡代理还是外地代理或者两者都是，而 KeyUsage 字段详细说明了密钥参数的用途。

为了产生一个用于认证服务的短期密钥，MoIPS 系统使用 Diffie-Hellman 算法。开发这样的短期密钥的主要设计目标如下。

- 由所有移动节点和移动代理使用。
- 不修改移动 IP 消息和扩展的格式。
- 不用于加密操作。
- 有力地保护主密钥。
- 与其他基于 Diffie-Hellman 的协议相关性低。

为了降低选择 IPSec 隧道过程中的通信开支，所有消息交换将作为移动 IP 认证控制消息的扩展来实现。进行隧道选择所需的消息交换的顺序如下。

① 移动节点根据代理请求和代理通告消息选择它与外地代理之间的 IPSec 隧道，还要根据它自己的安全策略选择 MN—HA 隧道。

② 移动节点把它的选择记录在一个隧道选择扩展(如图 9.21 所示)中与注册请求消息一起发送给外地代理。一旦收到该消息，外地代理经过检查之后转发给家乡代理。

③ 收到注册请求消息之后，家乡代理检查隧道选择并发送注册应答消息。当实现了所选择的分组隧道时，IPSec 数据认证和数据保密性服务使得移动节点可以像连接在家乡网络上一样，获得网络连接并进行安全通信。在该模型中采用了 ESP 的隧道模式。

下面是对该模型的几个假定。

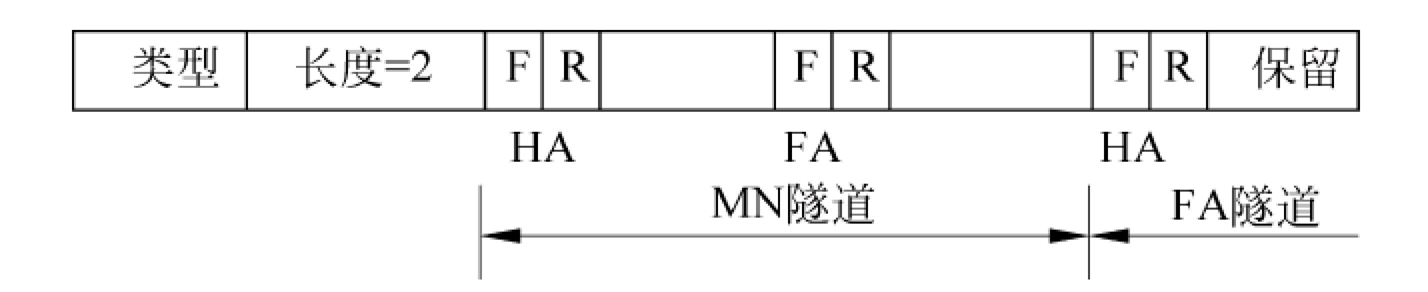


图 9.21 移动 IP 注册请求中的隧道选择扩展

- 为了更好地利用该框架，外地代理和家乡代理应该提供加解密和基于认证的分组过滤。
- 受防火墙保护的子网必须能够使距离移动节点最近的防火墙作为外地代理，而网络中所有其他防火墙应该允许 IPSec 分组通过。
- 受防火墙保护的子网必须能够使距离移动节点最近的防火墙作为家乡代理，而网络中所有其他防火墙应该允许 IPSec 分组通过。

可能选择的 IPSec 隧道包括 MN—CN、MN—HA、HA—FA 和 MN—FA。其中，MN—CN 是端到端的隧道，可以不考虑移动 IP 的存在。然而，当移动节点改变位置时，这些隧道可能被频繁地建立。其他的三种隧道中，MN—HA 隧道最有用，而 NN—FA 作用最小。

- HA—FA 隧道：家乡代理和外地代理之间的 MIP—IPSec。隧道的建立最简单，因为 IPSec 隧道很容易附加在现有的移动 IP 隧道之上。当它们用于提供数据认证和保密时，这些隧道在家乡网络和外地网络之间提供了一个 VPN 连接。
- MN—HA 隧道：提供了数据源认证和保密性，是最有用的隧道，因为它在移动节点和家乡网络之间提供了一条安全的通信路径。数据源认证防止了欺骗，而保密性避免了数据被窃听。建立 MN—HA 隧道的开销最大，因为它不是分组重定向机制的一部分，而且总是存在一个外地代理的干涉。这就导致了外地代理处的瓶颈，而且增加了可信任实体的数量。
- MN—FA 隧道：用于没有链路层保护的情况。它为移动节点提供了在外地网络上的数据保密性，以及 MN—HA 交换过程中的数据源认证。只有当移动节点选择使用外地代理的转交地址时才采用这种隧道。应该尽量避免采用这种隧道，减少不必要的开销。

MoIPS 系统提供了移动 IP 的管理消息的认证及重定向 IP 分组的安全隧道，在可扩展性方面和安全性方面确实具有一定的优势。但是由于该系统采用了公钥基础设施，安全隧道的建立涉及的协议包括 IPSec 和 IKE，这对于计算能力较低的移动节点和大规模网络中的移动代理意味着较高的数据延迟和网络负荷。

目前，尽管公钥算法能够提供足够好的安全性，但是关于在移动 IP 中使用公钥体制的问题还存在很大争议。因为在要求实时通信的条件下，在计算能力有限的移动设备中采用公钥算法必定引入相当大的延迟。但是，随着移动计算技术的不断提高，也许公钥体制很快

就能够在移动通信中得到广泛的应用。

9.6.3 移动 IPSec 方案

利用 IPSec 隧道可以增强移动 IP 的安全性，可以建立多种隧道，如移动节点—家乡代理隧道、家乡代理—外地代理隧道及外地代理—移动节点隧道等。

例如，瑞士伯尔尼大学提出了一种称为 SecMIP(secure mobile IP)的方案，以解决移动用户安全访问受防火墙保护的 VPN 的问题。如图 9.22 所示，该方案中的场景涉及一个专用网络和一个非军事区 DMZ，防火墙位于专用网络和非军事区之间，是进入企业专用网络的唯一入口，这样简化了安全管理问题。在该方案中，所有属于专用网络的移动节点和来自其他网络的访问节点都被置于 DMZ 中。因为属于企业的移动节点必须穿越防火墙来访问专用网络，它们必须使用 IPSec 来向防火墙认证其身份。

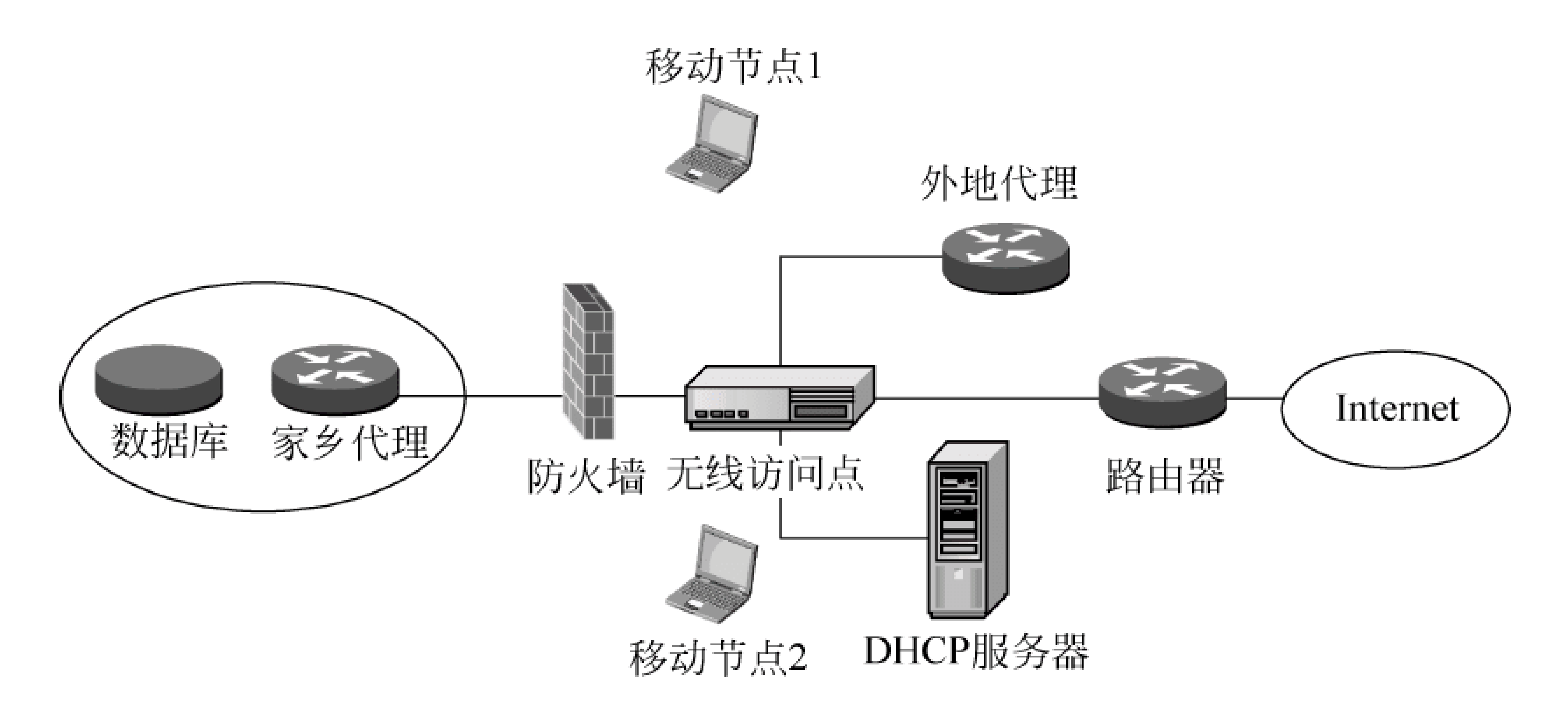


图 9.22　SecMIP 的配置场景

移动节点进入外地网络后，监听周期性地向非军事区发送外地代理通告消息。某些情况下移动节点可以发送代理请求消息来得到一个代理通告消息。一旦收到新的代理通告消息，移动节点得知它已经进入了一个新的区域，于是销毁所有旧的 IPSec 隧道。这时移动节点可以从一个 DHCP 服务器或者外地代理那里得到一个新的 IP 地址。该方案的作者并没有提到在移动节点和外地代理之间的任何认证过程。如果假定移动节点和外地代理之间没有认证，那么一个恶意的节点便有可能得到一个 IP 地址。尽管恶意节点不能成功地伪装成其他合法节点并得到发给它的数据分组，但是当恶意节点数量很多时，将导致外地网络 DHCP 服务器的 IP 地址的耗尽。

因为数据分组要穿过不安全的公众网络，合理的解决办法是在移动节点和家乡网络之间交换任何移动 IP 消息之前，在移动节点的转交地址和家乡网络的防火墙之间建立一个 IPSec 隧道。IPSec 隧道为所有移动 IP 注册过程中发送的 IP 分组提供认证性、完整性和保密性。接下来移动节点要将其当前的转交地址注册到它的家乡代理。家乡代理和移动节点

之间所有的移动 IP 注册消息通过 IPSec 隧道传输，因此不需要额外的认证和加密。

直到下一次的移动之前，移动节点可以和任何专用网络之内或之外的节点进行通信。出于安全的考虑，移动节点和通信对端之间的所有数据传输都要经过家乡代理。在不需要安全服务的情况下，移动节点可以直接和通信对端通信。经过加密和认证的移动 IP 分组由家乡防火墙解密并拆封后发送给家乡代理。家乡代理最终拆封这些移动 IP 分组，并转发给最终的接收者。

该方案能够允许移动节点访问受防火墙保护的 VPN 网络，它基于现有的标准，而且只需要对现有协议进行最少的修改。但是它的不足之处在于，由于移动节点始终不能真正回到家乡网络，只能在非军事区内与家乡网络内的节点通信，因此所有的数据分组都要进行严格的加密和认证，这将导致大量的 IP 分组频繁出入非军事区，给本来就已经繁忙的防火墙带来了额外的负担。而且大量的敏感数据在专用网络之外传输，也会带来安全上的隐患。另外，移动 IP 的注册消息封装在 IPSec 隧道中，多重封装也会带来协议上的复杂性。

9.6.4 穿越防火墙的 IP 移动方案

Stefan Mink 等人针对移动 IP 面对私有地址、防火墙、网络地址转换及服务质量时产生的问题，提出了一种无意识的透明穿越防火墙的 Internet 移动体系（firewall aware transparent Internet mobility architecture，FATIMA）。

FATIMA 的基本特征如下。

① 对于移动节点和通信对端的透明性。任何新的移动 IP 扩展对于这些节点必须是隐藏的。

② 关键性安全功能的集中化。这种性质是使用防火墙的主要好处。

③ 所有场合中的相互认证。这种认证可以防止伪造消息。

④ 高效的微移动支持。在同一网络的子网之间的移动能够比不同网络之间的移动更有效地被控制。

这种体系中，涉及的实体包括 FATIMA 网关、家乡/外地代理人（proxy）、外地代理和路由代理。FATIMA 网关是中心移动支持代理，它位于网络防火墙非军事区的堡垒主机中。所有的外地代理都由相对更简单的相应的代理人 FAP（FA proxy）来代替。FAP 不处理任何来自移动节点的注册请求消息，而只是将这些消息传给 FATIMA 网关。类似地，所有家乡代理由相应的代理人 HAP（HA proxy）来代替，只用于重定向数据分组，而所有的注册请求都由家乡网络的 FATIMA 网关来处理。在一个拥有众多子网的大型网络中，包括大量的 HAP 和 FAP，路由代理用于创建一个 FAI，IMA 网关路由器和代理人之间的等级体系。以上所有实体均采用 ESP 的隧道模式来路由从父节点到子节点的 IP 分组。

FATIMA 模型中数据分组的路由：在该模型中，当移动节点在外地网络时，任何发往移动节点的数据分组首先由 HAP 截获，并路由到外地网络的 FATIMA 网关路由器。然

而，当位于外地网络中的移动节点发送分组给另一个位于同一外地网络的移动节点时，拥有这两个节点的入口记录的中间路由代理能够直接将该分组转发给目标移动节点。由于分组不必始终通过家乡代理再返回移动节点，这样大大提高了效率。

当移动节点发送分组给位于同一外地网络中的固定主机，或者该固定主机发送分组给位于同一外地网络中的移动节点时，该分组将始终被转发到 FATIMA 网关路由器再路由到目标主机。即使在这种情况下，效率也得到了提升，因为分组并不经过家乡代理。在所有的实体之间都需要认证，即 FATIMA 网关路由器、路由代理和移动节点之间。在 FATIMA 中，确定了以下认证实例。

① 网络基础构架实体之间的相互认证。这主要是指 FATIMA 网关路由器，路由代理，以及属于同一网络的 FAP 和 HAP。这些认证通过使用 ESP 的隧道模式来实现。

② 移动节点和家乡网络基础构架之间的相互认证。为实现这种在移动节点和家乡网络 FATIMA 网关路由器之间的认证，在注册请求中要包含 MN—HA 认证扩展。该认证扩展很容易建立，因为移动节点和家乡网络 FATIMA 网关路由器都属于同一管理区域。

③ 家乡网络和外地网络基础构架之间的相互认证。这种相互认证，通过使用 HA—FA 认证扩展来实现。该扩展在采用 PKI 的情况下非常易于实现。然而，即使在没有 PKI 的情况下，由于每个网络中只有一个 FATIMA 网关路由器而有大量的外地代理和家乡代理，这种方法大大减少了家乡网络和外地网络之间需要的安全关联的数量。

④ 移动节点和外地网络基础构架之间的相互认证。这可以通过由家乡网络 FAI、IMA 网关路由器提供的认证来实现。如果家乡网络的 FATIMA 网关路由器认可外地网络路由器，也就隐含地为移动节点提供了对外地网络的认证。类似地，如果家乡网络的 FATIMA 网关路由器认可了移动节点，也就隐含地为外地网络提供了对移动节点的认证。

这种方案在安全性方面比较完善，在所有实体之间都提供了相互认证。但是，由于该方案中引入了较多的新的实体，明显地与现有的移动 IP 标准和基于 IPSec 的 VPN 网关不兼容，而且方案的复杂性较高，对于现有的系统升级需要的费用非常庞大。

思考题

1. 常用的增强无线局域网的安全技术有哪些？
2. 802.1x 认证协议的原理和实现方法是什么？
3. WAPI 的工作原理是什么？
4. 增强移动 IP 网络安全的措施和方法有哪些？

第10章

Web Service 与网格安全

10.1 Web Service 及其安全性概述

10.1.1 Web Service 简介

Web Service(Web 服务)是一种基于网络的,分布式、模块化的应用程序,具有面向服务的体系结构(service oriented architecture,SOA)的特性。Web Service 中,各种异构的应用程序或可执行代码的程序块被使用标准的语法(web service definition language,WSDL)进行描述,并使用一种通用的格式,即简单对象访问协议(simple object access protocol,SOAP)进行封装和传输,实现资源共享。这种标准化和通用性使得企业之间可以互相通信,并且不需要了解对方实际的 IT 设施和资源,从而实现异构资源的共享与整合。Web Service 是自包含、模块化的应用程序,可以在网络(通常通过 Web 方式)中被描述、发布、查找及调用。Web Service 也称为应用服务(application services),其体系结构如图 10.1 所示。

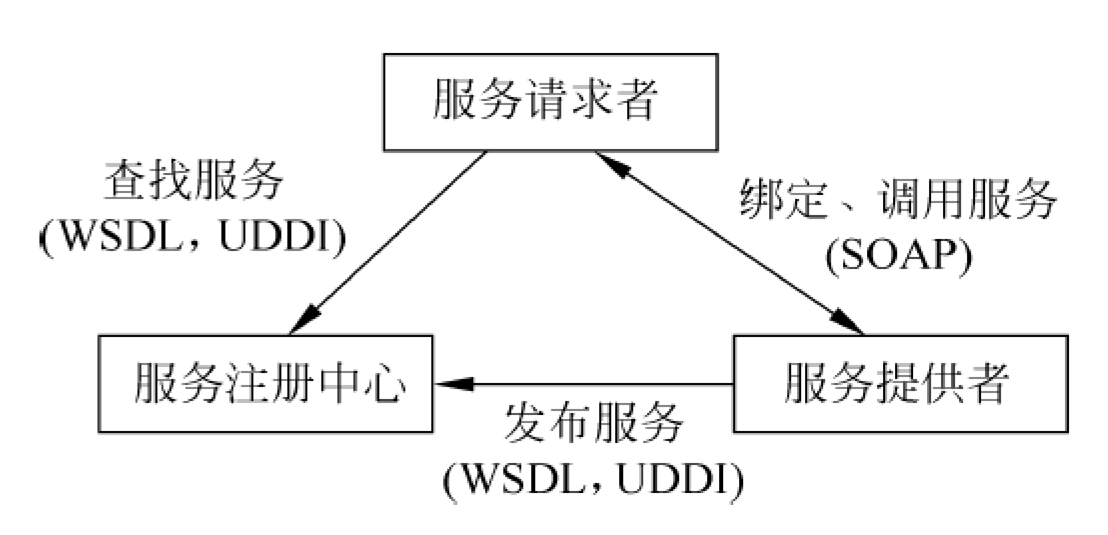


图 10.1　Web Service 体系结构

Web Service 包括三种主要角色:服务请求者、服务提供者及服务注册中心,相应的交互行为是发布服务、查找服务、绑定和调用服务。Web Service 使用的主要协议或语言包括 XML、SOAP、WDSL 和 UDDI(universal description discovery and integration)等。其中,XML 用来标记数据,为基本的数据描述语言;SOAP 用来传输数据;WSDL 用来描述服务,而 UDDI 用来注册服务。

服务提供者是服务的所有者,它通过标准、规范的基于 XML 的 WSDL 描述语言,描述与 Web Service 进行交互的具体细节,包括使用的传输协议、消息格式等。同时,服务提供

者的服务通过 UDDI 在服务注册中心进行发布。服务请求者是要求获得某些特定服务的实体，它们可以直接检索服务描述，也可以在服务注册中心查询所要求的服务类型。当发现合适的服务时，服务请求者可以通过 SOAP 绑定或调用服务提供者提供的服务。

SOAP 不是一个独立的网络通信协议，而是建立在 HTTP、SMTP 等协议基础上，通过这些协议进行 XML 格式的 SOAP 消息传送的，但同时 SOAP 也是独立于网络通信协议的，因为它不依赖于某一个具体的网络通信协议。

由于 WSDL 描述语言在描述 Web Service 时，隐藏了服务实现的细节，允许通过独立于服务实现，独立于软、硬件平台及编程语言的方式调用 Web Service。因此 Web Service 具有松散耦合性，特别适合于异构环境。

与现有的分布式应用技术相比，Web Service 具有如下优点。

① 它使用开放式标准（UDDI、XML、SOAP、WSDL），很好地解决了异构数据的表示、传输和查询的问题。

② 不依赖于二进制通信。传统的分布式技术，如 CORBA/IIOP、COM/DCOM 等均依赖于二进制通信，这使得通信过程常常被防火墙阻断。Web Service 使用 SOAP 协议和 XML 文本很好地解决了这一问题。

③ 具有跨平台的功能。传统的分布式技术通常情况下都会依赖于其设计的平台和设计语言，这就大大限制了用户对资源的使用。而 Web Service 允许任何可以创建 XML 文档并在 HTTP 上发送信息的程序设计语言与其交互。

④ 协议简单、易实现。传统的分布式技术一般包含很多内置的服务，如事务、安全性和加密，这些特性都增加了额外的开销，而且往往会产生不兼容的问题。而 Web Service 到目前为止还没有为任何高级别的服务提供和指定 API。这使得 Web Service 的实现变得非常简单。

由于具有诸多优点，使得 Web Service 的应用非常广泛，目前其应用领域大致可以分为如下 4 类。

① 面向商务的 Web Service。此类服务针对的是面向企业应用的服务。如企业内部 ERP 系统、企业间 RCM 系统等。它可以使公司内部的商务处理更加自动化。

② 面向消费者的 Web Service。这类服务主要向用户提供一些简单的功能。如允许用户在自己的客户机上集成网上购物 Web Service，查询商品价格、定购商品；集成医院 Web Service，查询病情、预定专家门诊等。这样不仅方便了消费者，也使整个过程自动化程度更高。

③ 面向系统的 Web Service。这类服务是将现有不同的应用程序无缝地结合起来。通常，不同的公司、企业会根据自身的特点和需求设计软件系统以协助管理。其中包含一些面向系统的程序如身份验证。如果将这些程序以 Web Service 的形式推广到 Internet 中，那么原先独立的公司、企业就可以使用同一个验证机制对 Web Service 用户进行身份认证了。

④ 面向设备的 Web Service。这类服务的使用终端一般是手持设备，如手机和日用家电等，这种服务使得智能化家电的广泛应用成为可能。

10.1.2 Web Service 的安全性需求

Web Service 的分布式、异构的本质及其开放性、跨平台和互操作性等特点使得其安全性比传统网络服务变得更加复杂。Web Service 的安全性需求具有如下特点。

(1) 端到端安全

传统的安全传输协议，如 SSL 和 IPSec 只能在点对点的情况下为传输层提供消息的完整性、机密性及消息来源的鉴别。由于 SOAP 消息可以由中介体接收并处理，所以即使通信双方的通信链路是可信任的，如果在中介体之间没有信任关联(trust association)，那么中间人可能会故意或通过在两个传输层的安全会话之间留出“空隙”来泄露信息，因此，端到端的安全难以保证。

(2) 传输独立性

Web Service 框架只定义了封装消息的格式，并没有给出传递消息的专用协议，它依靠和应用层协议的绑定来完成消息的传送任务，在传送消息的过程中可能会绑定不同的传输协议，所以当安全信息(如消息发送者的身份验证)需要被转移到消息路径中的下一个传输安全域时，可能会导致完整性方面的缺陷。

此外，虽然目前 Web Service 几乎都使用 HTTP 作为传输载体，但是以后的 Web Service 需要使用更可靠的消息发送架构，因此 Web Service 的安全不能依赖某一种底层网络协议的安全机制。

(3) 元素级安全性

XML 封装的数据是一种结构化的数据，包含了控制信息和消息内容两部分，其中消息的内容包括需要传送的各种层次化信息。如果将 XML 文档整篇加密，然后安全地发送给一个或多个接收方，可以通过 SSL 或 IPSec 实现，但它们无法实现对同一文档的不同部分进行不同处理，无法实现对不同元素组的授权访问。例如在进行商务交易时，商家可能需要知道顾客的名字和地址，但是不可以知道顾客使用的信用卡的情况；而在银行方面就需要知道用户信用卡的详细情况，而不需要知道他们采购商品的情况。显然，对于这种安全需求及应用的灵活性，SSL、TLS 和 IPSec 是无能为力的，而对不同对象采用不同的授权级别在 Web Service 中是必须的。

(4) 终端用户的间接访问

很多情形下，不是由终端用户直接访问 Web Service，而是由第三方代表终端用户访问 Web Service。Web Service 无法了解终端用户的实际身份，因此难以做出相应的授权决策。

例如，在图 10.2 中，终端用户通过浏览器访问一个商品 Web 站点，做商品预定。预定系统是一个 Web Service，商品 Web 站点通过 SOAP 协议访问此服务。用户在登录商品站

点时,Web 站点服务器可能已经对其进行了身份验证(例如采用用户名/密码的验证方法)。身份验证通过后,商品站点会向用户展示其商品,在确定用户希望预定何种商品后,Web 站点的 Web Service 服务请求者会向 Web Service 服务提供者发出预定请求。这时 Web Service 服务提供者只能看到商品 Web 站点,而看不到终端用户。即 WWW 服务终端用户的身份信息无法传递到预定系统 Web Service 的服务请求者和服务提供者之间的通信过程中。因此如何让 Web Service 验证终端用户的身份,防止非法入侵,是 Web Service 需要解决的关键安全问题之一。

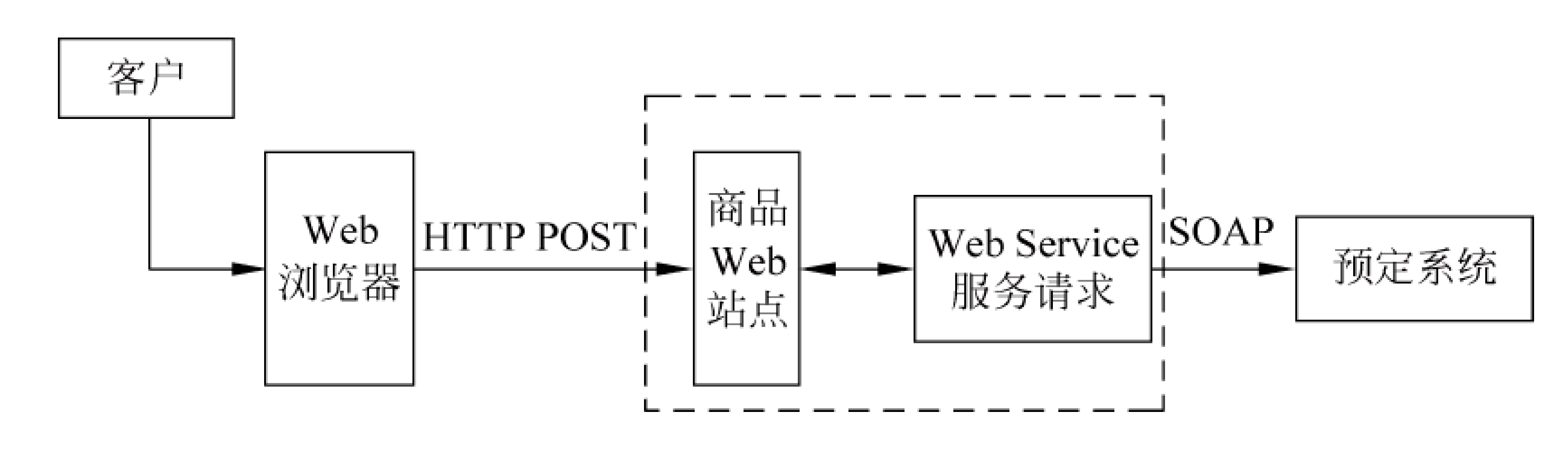


图 10.2　Web Service 安全性示例

为建立一个安全的 Web Service,通常需要从消息的机密性、完整性及访问控制等方面加以考虑。消息的机密性用于保证在 Web Service 的通信过程中,重要、私有的数据不被泄露,防止未经授权的第三方窃取信息。消息的完整性用于保证消息在传输过程中不被第三方篡改,确保消息在传输过程中是真实可靠的。访问控制用于防止第三方伪造相关信息(如用户身份等),非法入侵而破坏系统的可靠性、真实性。为此,已开发了许多增强 Web Service 安全性的行业规范,主要包括两类安全技术。

① 用 XML 格式表示安全数据的标准。这类规范包括 XML 加密、XML 签名等。

② 用于保护 SOAP 消息的安全规范架构,由 WS-Security 规范定义。其目标是允许应用程序安全地交换 SOAP 消息,通过整合 SOAP 模式中的特殊扩展来处理编码和证书。

10.2　Web Service 安全技术概述

如前所述,通过对 XML 及 SOAP 进行安全扩展可以增强 Web Service 的安全性,包括 XML 加密、XML 签名和 WS-Security 等。XML 加密实现交换信息的机密性,防止未经授权用户、实体或进程窃取信息;XML 签名保证消息的完整性和一致性,使未授权用户不能篡改或删除信息,并可提供消息的抗抵赖性;WS-Security 安全规范提供多种安全模型和加密技术来保护 Web Service 及 SOAP 消息的安全,并且可以把有关安全性的声明和消息关联起来。

10.2.1 XML 签名

XML 数字签名(XML digital signature)是由 W3C 和 IETF 共同提出的一个规范。XML 签名并不是执行数字签名的新方法,非对称加密和散列技术仍然用于执行实际的数字签名。XML 签名是一个用 XML 表示的数字签名,可以用来签名 XML 格式的数据。

XML 签名在 Web Service 中的作用是能够选择性地签名 XML 数据。例如,可以签名传递给 Web Service 方法的各种 SOAP 参数。如果 SOAP 请求通过中间人传递给目标 Web Service,那么 XML 签名可以确保端到端的消息完整性和不可否认性。XML 数字签名可以应用于任意的数字对象。在签名过程中,首先计算数字对象的消息摘要,然后将消息的散列值连同一些其他信息存放在一个元素中,最后将该元素使用私钥进行签名。

XML 签名的数据格式如下。

```
<Signature>
    <Signedlnfo>
        <CanonicalizationMethod/>
        <SignatureMethod>
            (<Reference(URI = )?>
            <Transforms>?
              <DigestMethod>
              <DigestValue>
            </Reference>) +
        </Signedlnfo>
        <SignatureValue>
        <Keylnfo>?
            <object> *
</Signature>
```

这个结构中,Signature 是签名的一个标志,它包含三个子元素,分别是 SignedInfo、SignatureValue 和 KeyInfo。

一个完整的 SignatureInfo 结构描述了产生 XML 签名需要的信息。SignatureInfo 又包括 CanonicalizationMethod 和 SignatureMethod 两个子元素。其中子元素 CanonicalizationMethod 表明在进行摘要之前用于对 SignedInfo 元素进行规范化的算法。规范化用于将不同的原始数据翻译为同一种形式(范式)。规范化算法非常重要,如果不使用规范化算法,直接使用摘要算法,那么两个逻辑上一致的 XML 文档可能会由于诸如空格、回车等格式上的差异而产生不同的摘要值,从而破坏了签名。

子元素 SignatureMethod 表明产生加密签名的方法,即用于将规范化后的 SignedInfo 进行签名成为 SignatureValue 的算法。它由报文摘要算法、密钥相关的算法和其他算法组

成。计算签名的算法名称同样也被签名，这样如果签名算法被篡改，也将导致验证的失败，从而防止攻击者将算法替换成更弱的算法来进行攻击。为了提高可交互性，W3C 工作组指定了一系列必须实现的签名算法，而使用何种算法可由签名者来决定。此外，SignatureMethod 还允许用户自定义算法。

SignatureMethod 元素中的 Reference 元素指明需要计算消息摘要的信息，它包含摘要算法 DigestMethod、摘要值 DigestValue，并且使用 URI 指明被签名的数据。一个空的 URI 表示包含这个 Signature 元素的整个 XML 文档将被签名。由于 XML 签名允许签名多个文档和签名一个文档的多个部分，Reference 子元素可以出现一次或多次。

Transforms 元素是可选项，它标识可选的一组处理过程，这些处理操作可以是规范化(canonicalization)、编/解码(包括压缩和解压缩)、XSLT 和 XPath。其中，XPath 操作允许签名者忽略原始 XML 文档中的一部分，因此这些忽略的部分可以进行修改而不影响签名的有效性。如果一个 XML 文档包含对自身的签名，这个方法就相当有用。如果 Transforms 元素没有出现，就直接对数据对象的内容进行消息摘要计算。

SignatureValue 元素包含签名值。而 KeyInfo 是一个可选项，用于表示密钥的信息。KeyInfo 元素指明用于验证签名的公钥，可以通过证书、公钥名或密钥交换算法指定。因为签名者的公钥信息可以通过上下文获得，所以 KeyInfo 是可选的。由于 KeyInfo 在 SignedInfo 外部，当签名者需要将公钥信息和签名绑定时，可以简单地将 KeyInfo 元素作为签名的一部分。

从上面对 XML 签名结构的分析中可以归纳出创建 XML 签名的步骤。

① 列出需要签名资源的 URI。

② 产生资源的消息摘要。

③ 将每个摘要和所使用的摘要算法、转换信息使用<Reference>元素进行描述，并将<Reference>元素中的内容与要使用的签名算法和规范化算法信息相结合，构造<SignedInfo>元素。

④ 规范<SignedInfo>数据，计算数据的消息摘要。

⑤ 产生消息的签名，将签名值插入到<SignatureValue>元素中。

⑥ 如果需要表示签名密钥的信息，可以创建<KeyInfo>元素。

⑦ 上述创建的所有元素都添加到<Signature>元素中。

相应地，根据 XML 签名的创建过程可以得出签名验证的方法。SignedInfo 的核心验证包括两个步骤：验证对 SignedInfo 的签名和验证 SignedInfo 中每个 Reference 的摘要。在进行签名验证时，首先进行引用验证，即接收端首先验证在每个<SignedInfo>元素中的<Reference>元素所包含的摘要。如果生成的摘要值和<DigestValue>元素中包含的规定摘要值不匹配，则验证失败。如果验证成功，则对<SignedInfo>元素中的加密签名进行签名验证。

10.2.2 XML 加密

XML 加密(XML Encryption)也是 W3C 提出的一个规范。它提供加密 XML 文档、XML 元素和 XML 元素内容等多种数据的方法,同时也提供利用 XML 格式表示加密数据的方法。XML 加密容易部署,而且能够选择性地加密 XML 数据。加密一个完整的 SOAP 消息会降低应用程序的性能,此外,如果 SOAP 消息必须包含足够的有用信息(例如路由信息),并且这些信息需要中间节点解析,那么也不能加密完整的 SOAP 消息。因此,选择性地加密 SOAP 消息中的数据是常用的方法。

XML 加密包括两个主要的元素:EncryptedData 和 EncryptedKey。

EncryptedKey 表示加密密钥的相关信息。该元素用于交换对称会话密钥,即 EncryptedKey 中包含的是加密后的对称密钥。EncryptedKey 元素中包含一个子元素 ReferenceList,虽然它不是必需的,但是非常重要,它能够让接收方了解使用这个密钥解密的数据部分的信息。

EncryptedData 元素包含加密内容及其相关信息(如加密类型、加密密钥和加密后的密文等)。XML 加密的结果也存放在 EncryptedData 元素中。该元素的 Type 属性用来表示加密的类型(如 XML 元素、XML 元素内容等)。例如,EncryptedDataType="http://www.w3.org/2008/04/xmlenc#Element"表明加密的是 XML 元素,如果将 Type 的末尾改为#Content,则表示加密的是元素内容。EncryptedData 元素中有一个必需的子元素 CipherData,它表示加密后的数据内容。

EycryptedData 元素可以通过如下结构表示。

```
<EncrptedData Id? Type? >
    <EncryptionMethod/>?
    <ds:KeyInfo>
        <EncryptedKey>?
        <AgreementMethod>?
        <ds:KeyName>?
        <ds:RetrievalMethod>?
        <ds: * >?
    </ds:KeyInfo>?
    <CipherData>
        <CipherValue>?
        <CipherReference    URI?>?
    </CipherData>
    <EncryptionProperties>?
</EncryptedData>
```

- EncryptedData：是 XML 加密中的核心元素。它不仅含有加密的数据，而且还可以用来替代被加密的元素，包括 XML 文档的根元素。EncryptedData 元素在子元素中包含或者通过 URI 指定了加密数据。当加密一个 XML 元素或元素内容时，EncryptedData 元素将在完成加密后的 XML 文档中取代相应的元素或元素内容。当加密整个 XML 文档时，EncryptedData 将成为新的 XML 文档的根元素。
- EncryptionMethod：为可选的元素，它描述了应用于被加密数据的加密算法。只有在接收方知道加密算法的情况下，这个元素才能默认，否则将导致解密失败。EncryptionMethod 中允许的子元素由它的 Algorithm 属性 URI 确定。
- ds:KeyInfo：描述解密 CipherData 的密钥信息，如果密钥能在上下文中规定，就没有必要在传送的 XML 文档中显式地表示。
- EncryptedKey：该元素用于向接收者传送加密密钥。它可以作为一个单独的 XML 文档置于应用文档中，或在 EncryptedData 中作为 ds:Keylnfo 的子元素出现（如上例中所示）。密钥被加密传送至接收者，当 EncryptedKey 被解密后，获得的密钥可以直接应用到 EncryptionMethed 指定的算法上。
- ds:RetrievalMethod：是 ds:KeyInfo 的子元素，可能出现多次。如果在 ds:KeyInfo 中出现了多个 ds:RetrievalMethod 实例，它指向的 EncryptedKey 中必须包含同样的密钥，这些密钥可能使用不同的方式加密，或者发送给不同的接收者。
- CipherData：是一个必须存在的元素。它在 CipherValue 元素中包含 Base64 编码的加密字节序列，或者在 CipherReference 中包含一个指向含有加密字节序列的外部地址的引用。
- CipherReference：该元素确定一个过程，用来获得加密的字节序列。在获取加密字节序列时，首先获取 CipherReference 中 URI 属性确定的外部资源，如果 CipherReference 中还包含可选的 Transforms 序列，则应用这些变换来获得加密数据。例如，如果加密数据存在于某一个 XML 文档中，就可以采用 Transforms 的 XPath 和 Base64 解码来获得该数据。
- EncryptionProperties：该元素中存放产生加密数据过程中的附加项信息，例如时间戳和序列号等。

10.2.3 Soap 消息安全保护

为了提高 Web 服务的安全能力，增强 Web Service 中 SOAP 消息的安全性，IBM、Microsoft 和 Verisign 等公司与一些组织机构合作制定了统一的 Web 服务安全规范，推出了 WS-Security（也称 Web 服务安全语言）等规范（版本为 WS-Security 1.1）。

结构化信息标准促进组织（organization for the advancement of structured infamation standards，OASIS）批准 WS-Security 1.1 成为其标准。WS-Security 规范主要解决终端用

户如何在 Web 服务之间安全传递 SOAP 消息的问题。通过 WS-Security,用户可以把消息完整性、机密性和身份验证嵌入 Web 服务应用之间的消息交换过程中。

WS-Security 详细描述了如何将安全性令牌(用于消息鉴别)附加到 SOAP 消息上,以及如何将 SOAP 消息与 XML 签名、加密规范相结合,起到保护 SOAP 消息的作用。在 WS-Security 没有规定所需要的安全性令牌的具体类型,这使得 Web Service 系统具有一定的可扩展性,可以适应各种认证和授权机制。

Ws-Security 标准的目的是确保 Web Service 应用处理消息的完整性及机密性,规定 Web Service 协议 SOAP 的扩展及消息头。在基于 WS-Security 扩展的 Web Service 中,SOAP 传输的不再是简单的 XML 文本消息,而是实现了 WS-Security 扩展的 XML 文件。WS-Security 中的安全性元素<wsse:Security>置于可信 SOAP 消息的 XML 文档的头部,是包含安全操作信息的顶级元素,它描述了接收方需要的有关安全性(如加密和鉴别)的信息。

WS-Security 本身并没有定义新的安全协议,而是利用现存的安全标准和技术实现安全的 Web 服务。它提供了一个可扩展的框架,用来在 SOAP 消息中嵌入安全机制,包括数字签名、消息摘要和数据加密等。这些安全服务信息是作为附加的控制信息以消息的形式传递的,不依赖于任何传输协议。因而 WS-Security 具有传输中立性。这种基于消息的安全模型如图 10.3 所示,通过对 SOAP 消息的认证(使用安全令牌)、加密和签名,消息级安全模型相对于传统平台的传输级(点到点)的安全模型而言,更适合于 Web Service 这样的异构环境,并且能够有效地防止消息在经过中间节点时遭到第三方的分析、更改和破坏。

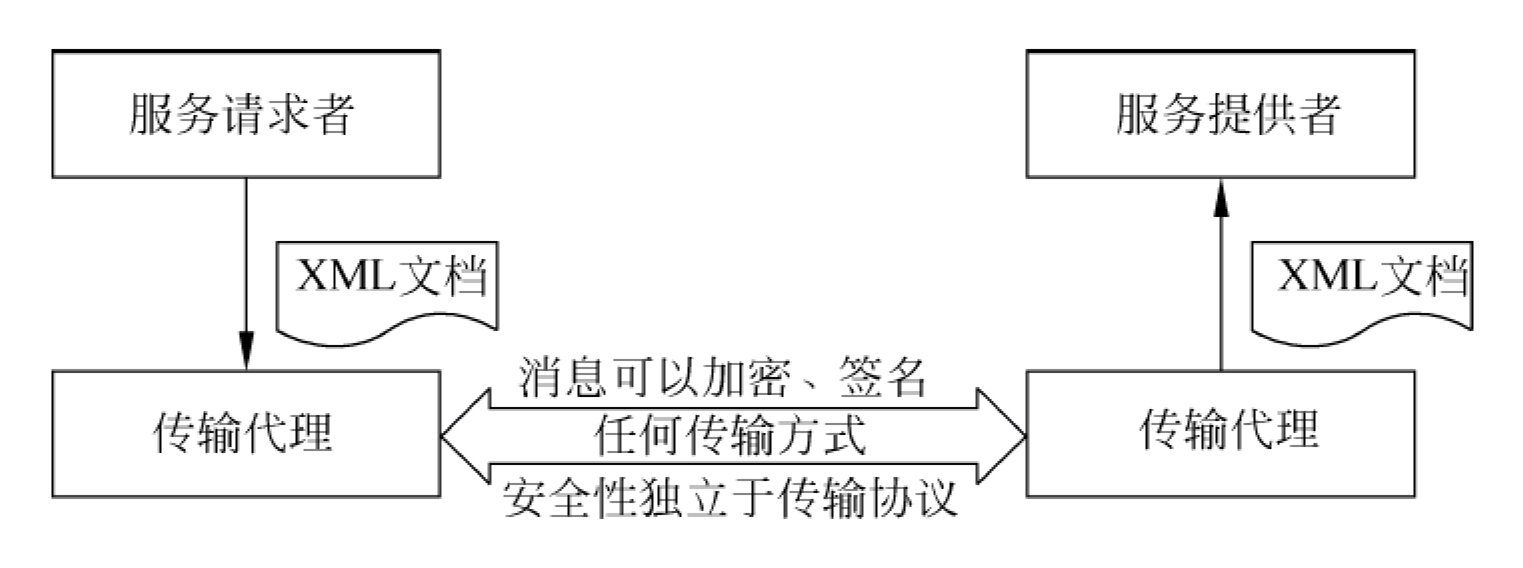


图 10.3 SOAP 消息安全模型

WS-Security 规范框架通过消息完整性、消息机密性和消息认证,提供 SOAP 消息传递的增强安全性,这些机制可以用于提供多种安全性模型和加密技术。

10.3 WS-Security

如图 10.4 所示,WS-Security 是一个安全框架(WS-Security framework),包括 WS-Security 核心规范及 XML 签名、XML 加密、用户名权标规范、X.509 权标规范、Kerberos 权标规范和 SAML 权标规范等。

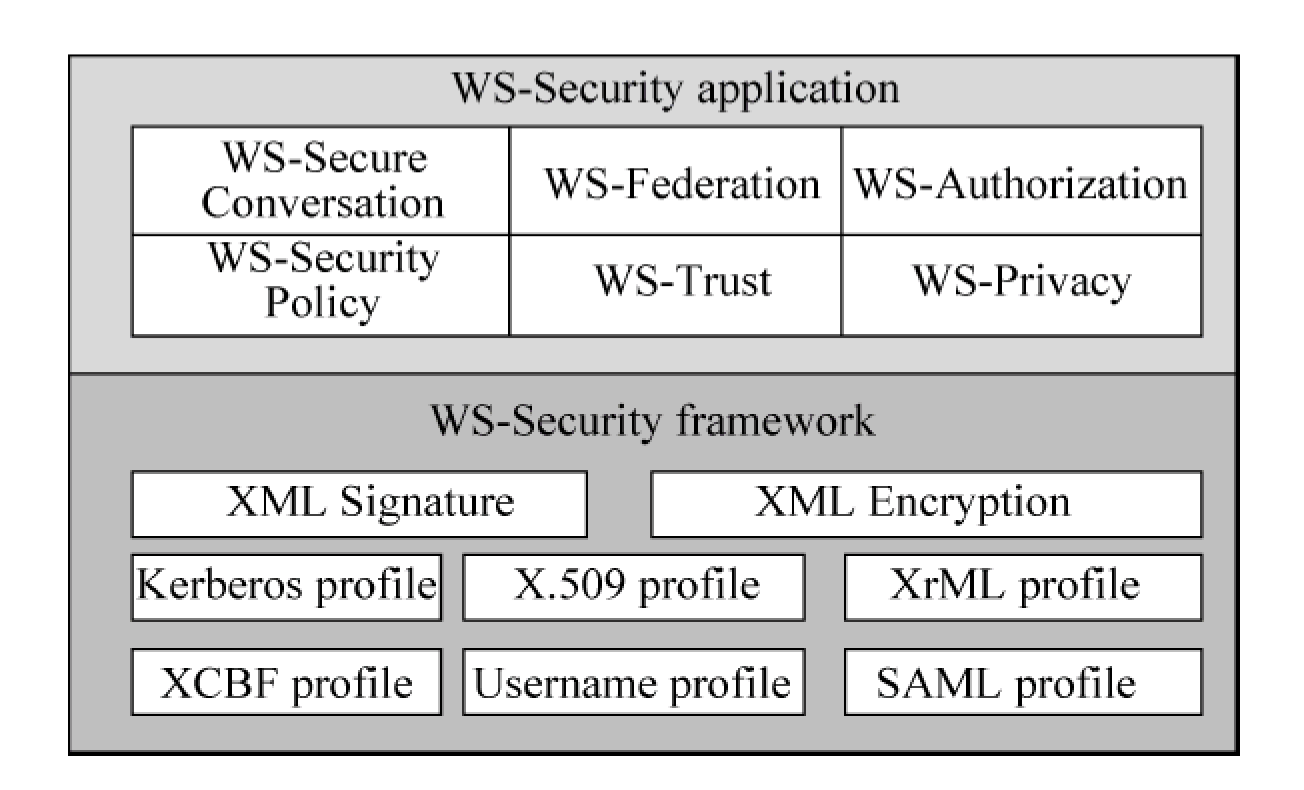


图 10.4　WS-Security 框架

在 WS-Security 安全框架之上，可以运行各种 Web Service 安全应用(WS-security application)，包括安全会话(WS-secure conversation)、身份联合(WS-federation)、授权(WS-authorization)、安全策略(WS-security policy)、信任(WS-trust)和私有性(WS-privacy)等。

其中，SAML 可用于实现单点登录，它定义了交换断言过程中所涉及的 XML 词汇、协议、传输绑定机制及配置文件。安全断言包括认证(authentication)、属性(attribute)和权限决策(authorization decisions)等。

作为服务器之间使用的认证协议，SAML 定义了交互过程中请求、响应的规范。例如，请求中主体查询、认证查询、属性查询和授权查询等。

基于 SAML 实现的安全应用包括如下方面。

① SUN 的 Liberty。

② Internet2 的用户身份联合认证 Shibboleth。此项目主要应用在校园内 Web 资源共享，以及校园间应用系统的用户身份联合认证。

③ XPOLA：基于能力的可扩展授权基础设施。在 XPOLA 中，安全能力(security capability)由策略文件和服务提供者对安全能力的签名组成。策略文件包括一个服务标识符、一个属主和与具体服务或资源对应的授权信息。每个安全能力都是一个 SAML 断言集合，对安全能力实施最小权利原则(principle of least authority，POLA)。

WS-Security 协议为 Security 元素提供了三类子元素，即安全性令牌 security token 类、签名类和加密类。WS-Security 特别为多安全性令牌、多信任域、多签名格式和多加密技术提供支持。规范提供了几种主要的安全机制：安全性令牌传播、消息完整性和消息机密性。

WS-Security 还提供关联安全性令牌和消息的通用安全机制。WS-Security 不需要特定类型的安全性令牌，它在设计上是可扩展的(例如支持多安全性令牌格式)。此外，WS-Security 还描述如何对二进制安全性令牌进行编码。此规范特别描述如何对 X.509 证书和 Kerberos 票据编码，以及如何加入难于理解的加密密钥，可以用于进一步描述消息中包

含的凭证特征的扩展性机制。

随着 WS-Security 规范的确定，各大软件厂商开始考虑为其产品提供和使用相同 Web 服务安全语言的接口和编程工具箱。Web 服务的开发者也将使用这些厂商提供的工具，加强开发 Web 服务的安全性。

10.3.1 WS-Security 消息模型

WS-Security 规范定义了一个抽象化安全服务的单一安全模型，其消息模型的结构如图 10.5 所示。

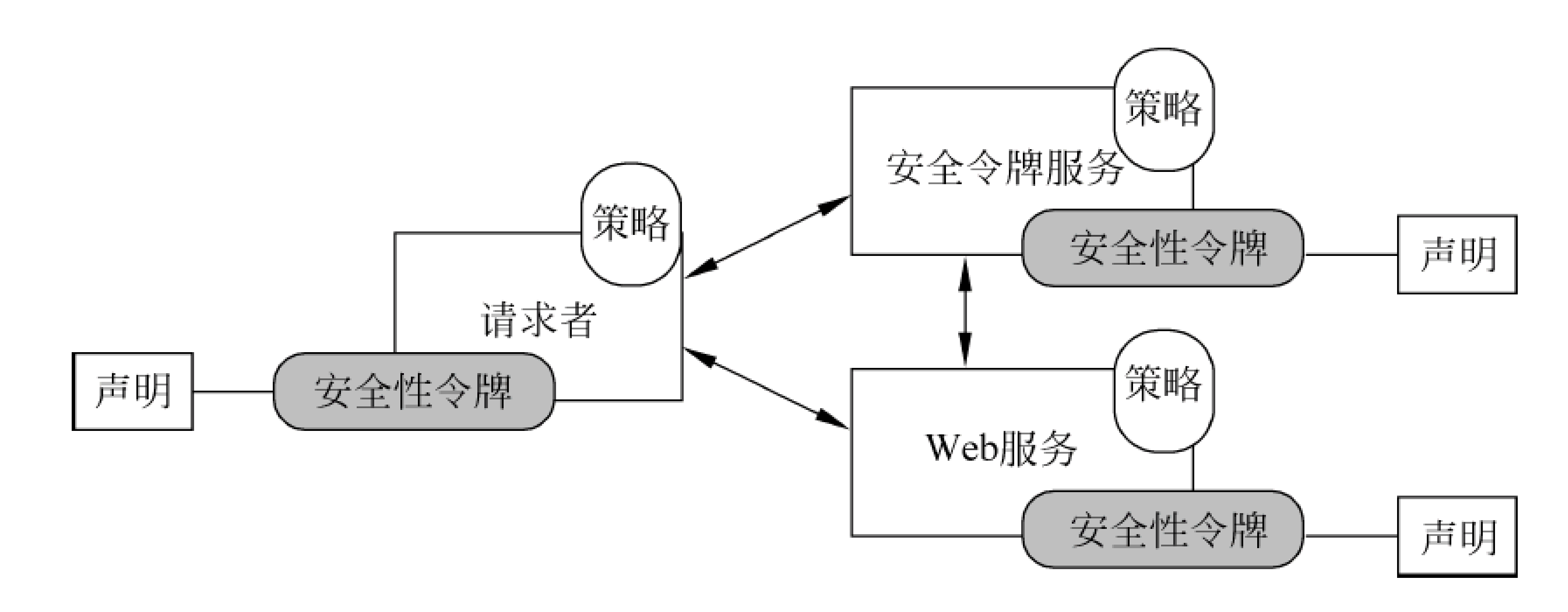


图 10.5 WS-Security 安全消息模型

从图中可以看出 WS-Security 的工作原理。首先，服务提供者根据自己的需要制定相关的安全策略，并要求所有请求其服务的服务请求者提供相应的声明和证书，其中声明包含主体的相关信息，如用户身份等。如果服务请求者无法提供相应的声明和证书，服务提供者可以拒绝提供服务。为了得到相应的声明和证书，服务请求者或服务提供者可以从第三方获取信任令牌。

第三方也是一个 Web Service 服务提供者，它所提供的服务称为安全令牌服务。由于第三方的服务提供者也具有自己的安全策略，因此不同安全域的服务提供者之间需要相互信任。为了取得这种信任关系，他们之间可以制定相互认可的安全策略。这样，即使服务需要跨越多个不同的安全域，WS-Security 也能够保证消息传递的安全性。当服务请求者具备相应的声明和证书之后，在请求服务时他们会将声明和证书随同 SOAP 消息一起发送给服务提供者。

10.3.2 WS-Security 基本语法要素

如前所述，WS-Security 提供身份鉴别、签名和加密等基本安全服务，相应地，在 WS-Security 中增加了安全性令牌(包括用户名安全令牌和二进制安全令牌)、XML 签名和

XML 加密等基本的数据元素和语法，以下描述 WS-Security 中的基本语法要素。

WS-Security 在 SOAP 头部定义了一个<wsse:security>元素，这个元素中包含与安全相关的信息，并使用 actor 属性指明消息的接收方。一个 SOAP 消息中可以包含多个< wsse:Security>元素，但是针对不同接收方的安全信息必须出现在不同的< wsse:Security>元素中。如果< wsse:Security>元素中不包含 actor 属性，则 Web 服务中的任何一方都可以使用< wsse:Security>中的信息，但是任何一方都不能将其删除。

此外，<wsse:Security>定义的块中还标识了发送方创建签名和加密的信息。WS-Security 规范并没有指定任何特定的次序来处理子元素(指<wsse:Security>块中的签名、加密子元素)。接收方应用程序可以根据自己的需要制定相应的策略来决定执行子元素的步骤。

安全性令牌是一组安全声明的集合。WS-Security 主要描述了两种安全性令牌，即用户名令牌(UsernameToken)和二进制安全令牌(BinarySecurityToken)。

(1) 用户名令牌

用户名令牌是一种提供用户名和可选密码信息的方法。该元素的示例 SOAP 消息如下。

```
<S:Envelope xmlns:S = "http://www.w3.org/2001/12/soap-envelope"
xmlns:wsse = "http://schemas.xmlsoap.org/ws/2002/04/secext">
  <S:Header>
     ⋮
    <wsse:Security>
      <wsse:UsernameToken>
          <wsse:Username>myname(/wsse:Username>
          <wsse:Password Type = "wsse:PasswordDigest">
              *************************
          </wsse:Password>
      </wsse:UsernameToken>
  </wsse:Security>
         ⋮
  <S:Header>
</S:Envelope>
```

消息中的用户名令牌元素中包括两个子元素：用户名和密码。其中密码元素包含一个名为 Type 的属性，该属性表明密码是以什么形式传输的。密码的传输形式有两种，即明文或摘要。

上述消息中的密码使用消息摘要的形式传输，这比使用明文传输密码要安全一些。使用这种方式，客户端首先创建密码的散列值发送给接收方，接收方接收消息后抽取<UsernameToken>元素，通过重新计算密码的散列来验证数据。

单独使用用户名/密码来进行身份验证的安全性不高，消息可能被第三方截获、重放。

因此，WS-Security 规范允许在用户名安全令牌中使用 nonce 或时间戳，以增强密码系统的安全性。

(2) 二进制安全令牌

二进制安全令牌使用 X.509 证书、kerberos 票据等非 XML 安全令牌进行身份认证。一个二进制安全令牌使用 BinarySecurityToken 元素表示，通常使用 ValueType 和 EncodingType 两种属性来说明。其中 ValueType 属性表明使用的令牌类型，在 WS-Security 中支持三种属性：用于 X.509v3 数字证书的 wsse:X509v3、用于 Kerberos 票据的 wsse:Kerberosv5ST 及用于 Kerberos 授权票据的 wsse:Kerberosv5TGT。EncodingType 属性表明安全性令牌的编码方式。

下面给出一段使用 X.509 证书的 SOAP 消息的示例。

```
<wsse:BinarySecurityToken
      ValueType = "wsse: X509v3"
         EncodingType = "wsse:Base64Binary"
         ID = "SecurityToken-XYZ"> ***** …
</wsse:BinarySecurityToken>
```

上述消息中使用 Base64 编码的二进制安全令牌 X.509 证书作为身份认证的方法。

当使用 Kerberos 安全令牌时，该令牌中包含了服务请求者向 Web 服务提供者证明身份及服务提供者向服务请求者证明身份的机制。Kerberos 票据只适用于访问一个 Web 服务，并用于验证服务请求者的身份。当在 SOAP 消息中提供 Kerberos 票据时，需要将该数据加密复制到消息中。

(3) 签名

在 WS-Security 规范中，<ds:Signature>安全元素的含义和上节的 XML 签名的含义相同，WS-Security 规范允许使用 XML 签名保护 SOAP 消息。

WS-Security 规范中 XML 签名通常会与安全性令牌结合使用，<ds:KeyInfo>元素中添加一个安全令牌引用元素<wsse: SecurityTokenReference>，用于指明所使用的安全性令牌。

(4) 加密

WS-Security 规范利用 XML 加密标准实现对消息的主体块、报头块，以及任意子结构和附件组合进行加密，并且在使用加密元素时，WS-Security 在 XML 加密规范的基础上进行了一定的修改。例如，XML 加密规范在<xenc:EncryptedKey>中使用<xenc:ReferenceList>元素，以表明所有被引用的<xenc:EncryptedData>元素都使用同一个密钥加密。同时，在 WS-Security 规范中允许同一个<xenc:ReferenceList>引用的<xenc:EncryptedData>元素在不同步骤中加密。因而可以使用不同的加密密钥，每个加密密钥都可以在<xenc:EncryptedData>内的<ds:KeyInfo>中指定。

和 XML 签名类似，如果加密是基于附加的安全性令牌的，那么就向<ds:KeyInfo>元

素添加一个<wsse:SecurityTokenReference〉子元素,以便于定位。

10.3.3 WS-Security 安全令牌信任机制

WS-Security 规范可以根据不同的应用需求采用多种安全性令牌信任机制来实现对 Web 服务的保护,下面介绍几种不同类型的安全令牌信任机制。

(1) 直接信任机制

直接信任包括两种类型:一种是用户名/密码的直接信任。即客户机(web service 服务请求者)使用 SOAP 协议向服务器(web service 服务提供者)发送请求,其中包含一个用户名/密码安全性令牌,服务器通过认证用户名/密码来判断用户身份,验证通过则处理请求并返回结果。

另一种信任是使用安全性令牌和签名的直接信任,即 Web 服务直接信任用户的安全性令牌(指服务提供者和服务请求者双方已经使用某种机制建立了 Web 服务对安全性令牌的信任)。在运行过程中,服务器会审核并评估安全性令牌,同时,为了验证身份,服务请求者会对安全性令牌进行签名。服务请求者将签名的安全性令牌包括在请求消息中,并提供与安全性令牌关联的密钥的所有权证明。服务提供者可以根据安全性令牌的签名验证服务请求者的身份,并将认证和请求处理结果返回给服务请求者。

使用直接信任机制时,安全性令牌直接由服务请求者发送至服务提供者,其模型如图 10.6 所示。

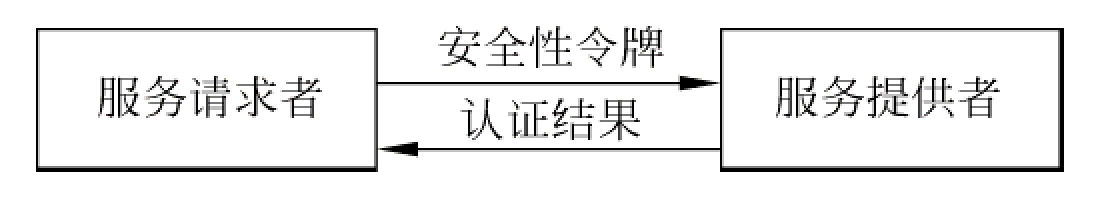


图 10.6　直接信任机制

(2) 间接获取令牌机制

这种信任模型需要使用第三方的安全性令牌服务。使用这种信任模型的服务请求者没有直接向服务提供者给出其安全性令牌,而是给出一个安全性令牌的引用。因此安全性令牌不是作为消息的一部分传递的,而是提供了一个用于定位和获取令牌的引用。

当请求者向服务提供者发送请求时,把对安全性令牌的引用包括在请求消息中,并以 XML 签名的形式提供所有权的证明。Web 服务利用所提供的信息从安全性令牌服务取得安全性令牌,并以此进行请求者身份验证。Web 服务处理完成后向服务请求者返回处理结果。如果使用这种方案,SOAP 消息的报头会添加如下安全性令牌。

```
<wsse:SecurityTokenReference wsu:Id = "…"
    <wsse:ReferenceURI = "…"/>
</wsse:SecurityTokenReference>
```

服务请求者在收到包括以上信息的 SOAP 消息后，会根据引用中指定的证书 URI，到相应的安全性令牌服务处取得证书或票据。

间接获取令牌机制的信任模型如图 10.7 所示。

(3) 签发令牌机制

签发令牌机制与间接获取令牌机制类似，但是该机制中，安全性令牌通过请求方，而不是服务方获取，即安全性令牌服务向服务请求者签发安全性令牌。当服务请求者获取了令牌后就可以多次使用这个令牌，以取得相应的 Web 服务。使用这种方案，SOAP 消息中的安全性令牌形式可以是上述提到的任意一种。其模型如图 10.8 所示。

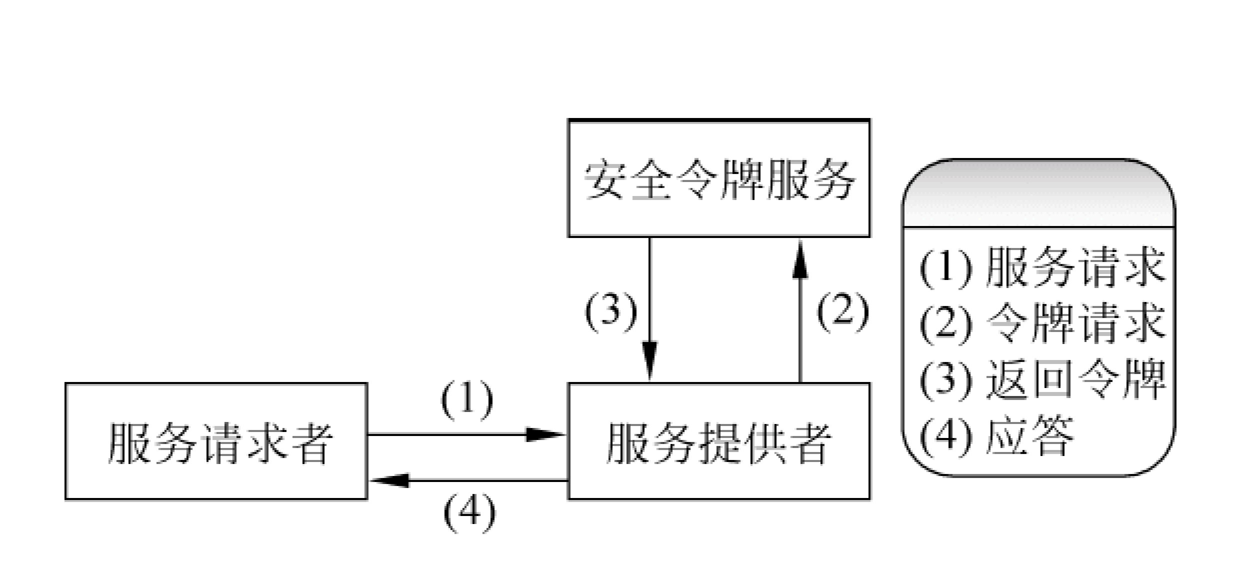


图 10.7 间接获取令牌机制

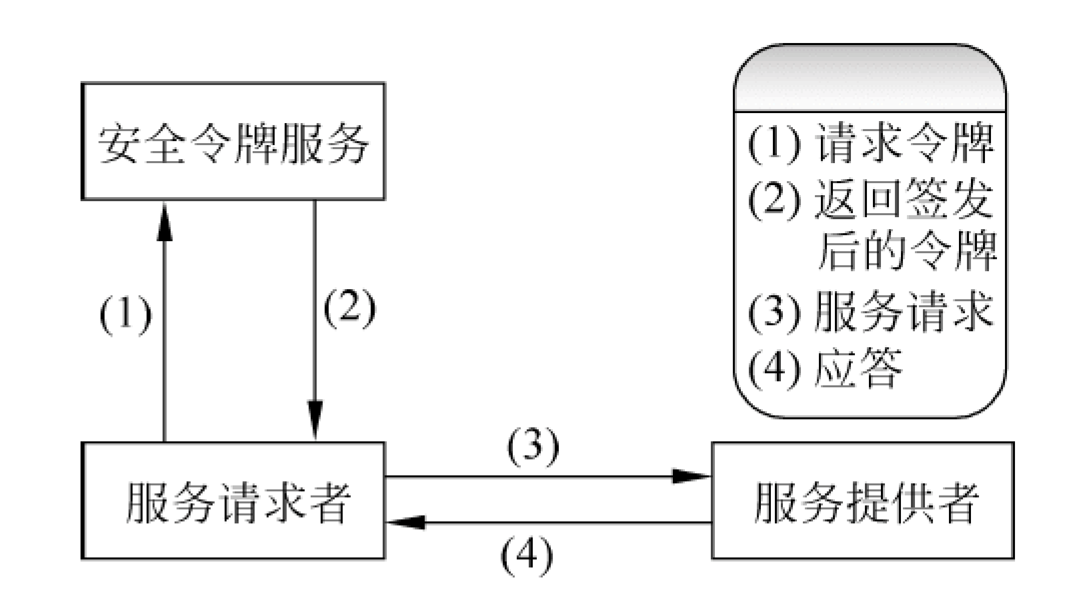


图 10.8 签发令牌机制

10.4 网格及其安全性概述

10.4.1 网格体系结构及其特性

网格可以将地理上分布、异构的高性能计算机、数据服务器和各种应用系统等通过高速网络连接起来，实现资源的全面连通和有效组织。

网格体系结构是划分网格系统基本组件、指定系统组件的目的与功能及说明组件之间如何相互作用的技术。当前，比较重要的网格体系结构有两个，一个是 Foster 等人在早些时候提出的五层沙漏结构；另一个是在考虑到 Web 技术的发展和影响后，结合 Web Service 提出的开放网格服务体系结构（open grid service architecture，OGSA）。

1. 五层沙漏结构

如图 10.9 所示，五层沙漏模型是以协议为中心的层次结构，以沙漏模型的原则刻画了网格的体系结构。

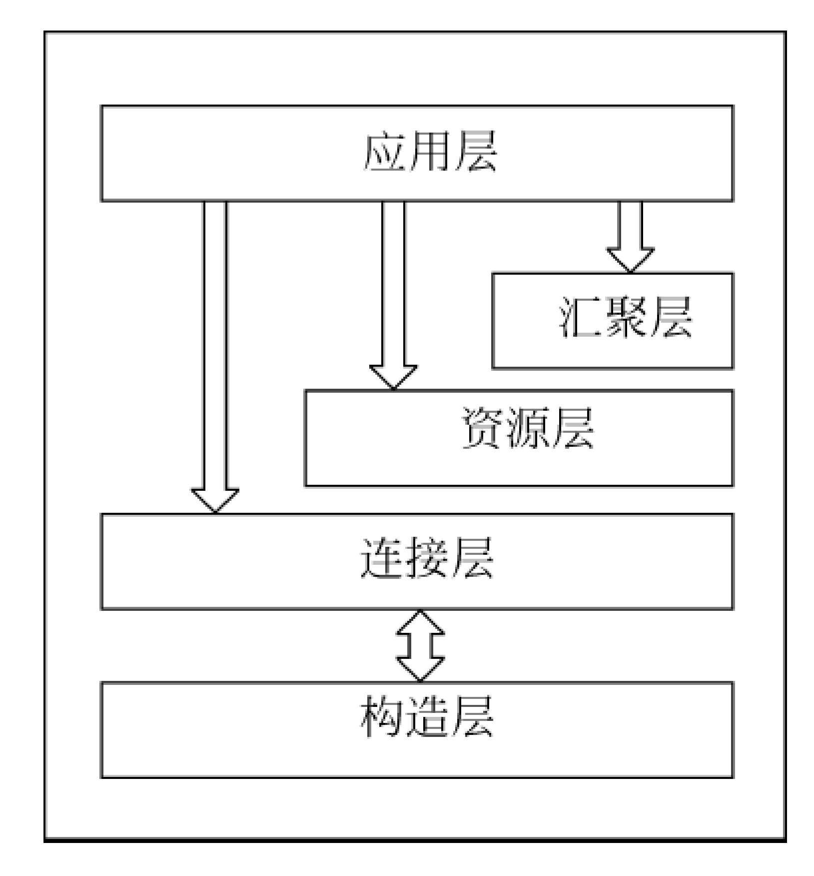


图 10.9 网格的五层沙漏结构

(1) 构造层(fabric)

网格构造层的基本功能是控制局部资源。构造层的资源是非常广泛的,可以是计算资源、存储系统、目录、网络资源和传感器等。构造层网格组件实现对本地特定资源的访问,向上提供访问这些资源的接口。因此在构造层各种功能实现上,有紧密和互相依赖的关联,还有对共享的支持。构造层实现的基本功能包括资源查询、控制服务质量的资源管理能力等。

(2) 连接层(connectivity)

支持便利安全的通信。连接层的基本功能是保证构造层的资源实体间相互通信的便利和安全。在这一层,网格定义了核心的网络事务处理所需要的通信和认证协议。通信协议允许在构造层资源之间交换数据,建立在通信服务之上的认证协议,以识别用户和资源。通信的必要条件包括传输、路由和命名等功能。连接层使用 TCP/IP 协议中现有的通信协议,如 IP、ICMP、TCP 与 UDP,以及应用层的 DNS、OSPF 和 RSVP 等。

(3) 资源层(resource)

资源层建立在连接层的通信和认证协议之上,用来共享单一的资源。资源层定义的协议包括安全连接、初始化、监视和控制、审计、计费等。资源层的协议实现调用构造层的功能以访问和控制本地资源。资源层最重要的两个协议是信息协议和管理协议,前者用于获得关于资源结构和状态的信息,后者用来协商对共享资源的访问。

(4) 汇聚层(collective)

汇聚层的基本功能是协调多个资源的共享,实现虚拟组织(virtual organization,VO)。汇聚层组件建立在资源层和连接层形成的协议之上,能够在不对资源实施新的要求的情况下实现广泛和多样化的共享行为,如目录服务、协同分配、调度和代理服务等。

(5) 应用层(application)

虚拟组织中的所有用户应用构成了网格的应用层,它调用下层的服务来构造网格应用。从网格应用开发者的角度来看,下面各层的协议和服务都提供了相应的 API 和 SDK,使得用户可以很容易地构建网格应用。

以资源共享协议和资源间的通信协议为核心,网格环境实现了广域范围内的资源共享和协同工作,将面向 Internet 的计算推进到了一个新的阶段。计算网格体系结构中的连接层、资源层和汇聚层的功能需要由架构在资源层之上、应用层之下的网格中间件实现。

2. 开放网格服务体系结构

OGSA 是五层沙漏结构之后最重要的网格体系结构,它基于面向服务体系结构。OGSA 中的服务是一种通过信息交换来提供给客户某种能力的实体,服务可定义为导致服务执行某些操作的特定信息交换的序列。只按照信息交换来定义服务操作,给如何实现服务及定义服务带来了极大的灵活性。在面向服务的体系结构中,内部实体都是服务,因此任何对体系结构来说可见的操作都是消息交换的结果。

下面三个特征强调了服务概念的一般性和应用的广泛性。服务也包括低级的资源管理

功能到高级的系统监控功能。

① 存储服务可以提供操作来存储和检索数据，预留空间，监控存储服务的状况，并查询和定义服务访问政策，以决定哪些实体能够访问服务。

② 数据传输服务提供操作，以将数据从一个存储服务迁移到另一个存储服务，对传输状况进行监控和管理，并查询、定义传输请求优先级排序的策略。

③ 故障处理服务可以监控其他各种服务的状况，例如存储服务和数据传输服务，并提供操作使得其他实体获得与错误有关的通知，以及查询和定义通知策略。

设计 OGSA 的一个目标是使得服务能以标准方式表示而不依赖于上下文，这样可以简化应用设计并有利于代码重用。为了实现行为重用，需要把操作组合起来形成服务接口，然后接口也可以组合起来规定期望行为的服务。OGSA 的第二个设计目标是实现服务组合。

如图 10.10 所示，OGSA 包括开放网格服务基础结构、OGSA 服务和 OGSA 模式三个主要组件。OGSA 是构建在 Web 服务之上的，但是现有的 Web 服务标准不能解决有关基本服务语义相关的问题，例如服务是怎样创建的，存活多久，如何处理错误及怎样管理长期状态等。这些服务语义和其他重要的服务行为必须予以标准化，以便使服务虚拟化和实现服务间的互操作。通过开放式网格服务基础结构的核心接口集可以解决这些问题。符合 OGSI 标准的 Web 服务就称为网格服务。

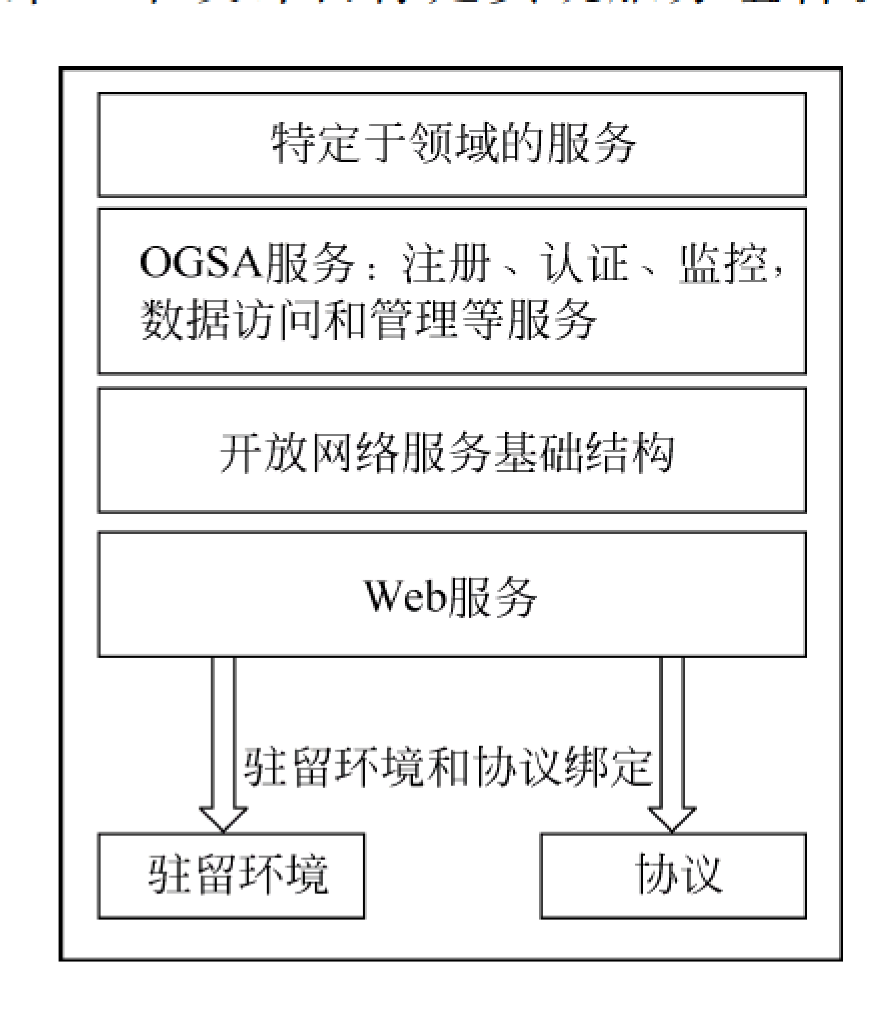


图 10.10 OGSA 的组件

OGSI 为分布式系统定义了基本的构造块，包括描述和发现服务属性、创建服务实例、管理服务生命期、管理服务组，以及发布和订阅服务的标准接口和相关行为。但是，OGSI 并没有定义创建大规模系统时所需要的所有组成成分。

此外，OGSI 还需要处理许多其他问题，例如，如何建立身份识别及协商认证；如何发现服务；如何协商和监控服务级协议及如何监控和管理服务集等。在这些区域内如果没有一个标准，将会很难以一个标准的样式建造大规模系统以实现代码重用和组件互操作。因此，OGSA 必须在这些区域及相关区域定义附加服务。可操作不仅与一种公共语言有关，而且还需要一组能够描述具有共同兴趣对象的公共词汇。因此，OGSA 必须定义标准模式来描述网格公共实体的属性。

3. 网格的特性

由于网格建立在一个开放的网络环境中（如 Internet），这种环境可能会遭受来自内部或外部的安全威胁。这些威胁可能源自用户的误操作，也可能是针对系统脆弱性的攻击，它们往往会带来严重的后果。尤其在网格技术由应用研究转向商业服务后，提高网格环境的

安全性问题显得尤为重要。

网格的诸多特性可能导致网格环境中的安全问题,这些特性如下。

① 参与网格的用户与资源的数目非常巨大,而且经常动态变化其参与状态。

② 应用程序可能在运行时动态请求、访问与释放资源,亦即应用程序由动态覆盖在某个资源集合上的不定个数的进程所组成。

③ 应用程序各个进程之间会动态建立和销毁信道,并可能使用不同的通信机制进行通信。

④ 每个网格节点,甚至每个资源或资源组都会有各自的访问控制策略(即自治性),而外界或应用很难,也不应该对其资源访问过程进行过多的干涉。

⑤ 为了记账和访问控制,一个用户在不同的站点可能有不同的名字空间、证书或账号。

⑥ 资源和用户可属于多个组织。

总之,网格的安全模型必须建立在一个动态、需要协调不同访问控制策略和不同安全互操作的环境中。

4. Globus

Globus 项目是目前最有影响力的与网格计算相关的项目之一,是来自世界各地关注网格技术的研究人员和开发人员共同努力的成果。它围绕 4 种主要活动来组织:研究、软件工具、实验平台和应用程序。Globus 对资源管理安全、信息服务及数据管理等网格计算的关键技术进行研究,开发可以在各种平台上运行的网格计算工具软件,帮助规划和组建大型的网格实验平台,开发适合大型网格系统运行的大型应用程序。Globus 工具包是 Globus 最重要的实践成果,它是一个开放源码的关键网格协议的参考实现。该工具包基于开放结构、开放服务资源和软件库,并支持网格和网格应用,致力于解决安全、信息发现、资源管理、数据管理和通信错误诊断等问题。目前,Globus 的技术已在 NASA 网格(NASA IPG)、欧洲数据网格(DataGrid)和美国国家技术网格(NTG)等多个项目中得到应用。Globus 的网格计算协议是建立在因特网协议之上的,以因特网协议中的通信、路由和名字解析等服务为基础。Globus 的协议分为 5 层:构造层、连接层、资源层、汇聚层和应用层。上层协议可调用下层协议的服务。

Globus 工具包包括网格安全、网格信息获取与分布、网格资源管理、网格数据管理及网格远程传输等内容,这些都是网格开发中的关键技术和必须解决的重要问题。Globus 网格计算环境中,所有可用于共享的主体都是资源,如计算机、高性能网络设备、仪器、大容量的存储设备、各种科学数据及各种软件等是资源,分布式文件系统、数据库缓冲池等也可以理解为资源。对于共享而言,有价值的不是设备本身而是实体的接口或界面。在 Globus 看来,大型应用项目应该由许多组织协同完成,它们形成一个“虚拟组织”,各组织拥有的计算资源在虚拟组织里共享,协同完成任务。Globus 并没有取代现有技术,而是在现有技术之上建立更高层次的资源共享和协同。

10.4.2 网格环境中的安全挑战

网格环境建立在原有的各类系统之上，并且需要能把它们整合起来，协同运作。所以无论服务在何处运行，它的安全系统都必须提供一个统一的解决方案，接口必须是抽象的，以便提供一个可扩展的结构。具有不同安全机制和策略的，在不同虚拟组织中的服务主机可以互相调用。网格安全应该能够综合利用现有系统和技术，并且具备支持互操作的协议，以及建立交互主机环境间的信任关系等多种安全服务能力。

这些要求之间的依赖关系如图 10.11 所示，任何一个使用联合证书来达到互操作目的的解决方案都会依赖于定义在参与域中的信任模型，定义一个独立的信任模型是实现互操作的基础。同样，统一形式的方案需要一个关系到互操作的信任层。

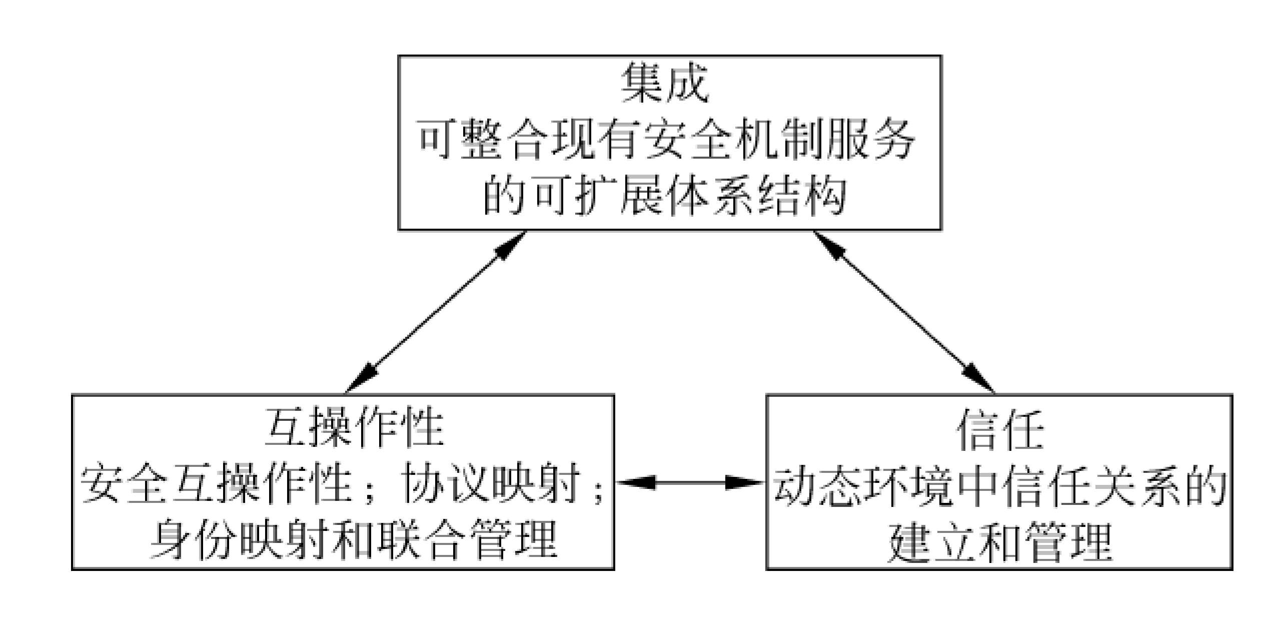


图 10.11 网格环境中安全挑战的分类

下面分析网格环境中的安全挑战。

1. 技术综合挑战

不论是出于技术和可编程角度的原因还是其他原因，期望使用一种安全技术来解决所有网格计算的安全问题是不现实的，原先的安全基础设施不可能在短时间内被取代。例如，网格环境中每个域可能都有一个或多个注册功能来维护用户账号。这些注册机构不太可能把用户信息共享给其他组织和域，同样，现有环境中被认为安全可靠的认证机制也会继续使用。为了每个域继续拥有，管理和支持自己原有的授权设施，单一的模型和机制无法为网格环境所接收。这样，网格安全体系结构必须是兼容各种实现，能够在不同的已有安全机制（Kerberos、PKI 等）上被实例化，必须有可扩展性以适应新的安全服务，能够兼容现存的安全服务。

2. 互操作性挑战

跨越多个域和主机环境的服务需要互相通信、协同工作，这样就必须引入多个层面上的交互性问题。

在协议层,需要能够在域间交换信息。

在策略层,参与安全对话的各方能指定任何策略,并且不同的策略能够被对方理解,这样各方就可以建立一个安全通信的通道,实现安全上下文的交互认证。

此外,对于用户身份的问题,网格需要能够识别来自本地域之外的其他域的用户,即建立跨域的信任关系。不考虑认证和授权模型,这种信任关系可以是基于组、基于身份、基于属性的。比较理想的情况是一个身份能够跨所有参与的域,但这不易实现。要使得能够在一个安全环境中跨域调用,身份或者证书的映射可以达到这个目的,映射可以通过代理服务器上的会话或扮演代理的信任中介来实现。

3. 信任关系的挑战

网格服务要求能跨多个安全域,这种域间的信任关系在端到端的交互中扮演了一个非常重要的角色。客户端必须知道服务的访问要求,以及如何安全地访问它。在网格环境中,虚拟组织的动态与分布式的特性是信任关系建立的最大障碍,每个会话中的信任关系在每次请求的时候必须进行动态评估。网格动态的特性使得这种信任关系难以建立,参与的域可能有着不同的安全基础设施,所以通过多种形式的安全机制来建立信任关系是必要的。

虚拟组织必须支持动态的、用户控制的部署和动态服务的管理。用户创建动态服务需要解决如下安全问题。

- 身份和授权:动态服务创建必须在受控的情况下实施。
- 策略执行:用户需要为他们创建的服务建立访问控制策略,这些策略必须限定在服务提供者的本地策略范围内。
- 安全层发现:用户可能希望在主机环境上建立安全层,这些信息必须能够被网格安全基础设施发现,如病毒监测、防火墙和 VPN 等。
- 策略组合:当一个服务实例被创建时,它的安全策略能够动态地合成。这些需要合成的策略包括资源拥有者的策略、请求者的安全策略和 VO 中这个实体的成员资格赋予它的策略等。
- 委托:在网格计算中,用户常常委托动态服务去执行相关操作,动态服务需要以用户的身份执行任务而不需要用户的直接干预,网格应该能够降低这种“委托”产生的风险。例如,一个计算任务需要访问不同资源上的数据,而这些资源所在的虚拟组织和任务所在的虚拟组织没有直接信任关系,那么用户需要委托动态服务代表自己去访问所需资源,如果这种权限的“委托”通过委托证书实现,则如何最小化用户委托给动态服务的证书的危险,以及控制任务完成前“委托证书”过期等应该成为重点考虑的安全问题。

10.4.3 网格的安全性需求及其安全架构

从用户的角度出发，网格安全面临的最基本的问题是如何在网格这样的复杂环境中提供给用户简单易用的安全功能。比如单点登录，一个用户只需要在提交网格计算任务前进行一次认证，以后在任务运行时，安全机制就可以在任务申请其他资源时进行自动认证，不需要用户再次参与认证过程。此外，用户证书和私有密钥必须得到安全而灵活的保护，使用户在任何地方都可以安全地使用自己的证书和密钥。最后，网格的安全机制应该对用户透明。

资源提供者则更关心如何控制和保护自己的资源。比如，虽然资源是提供给网格虚拟组织共享的，但是资源提供者有能力决定资源的本地访问控制策略，从而改变资源在虚拟组织中的访问控制。另外，本地安全系统和虚拟组织的协调和整合也是一个重要问题。取代或是修改这些安全系统是不现实的，只能通过映射和代理机制予以解决。

对于开发者而言，他们需要的是一个灵活和功能丰富的函数库，以方便地实现认证，灵活地实现消息保护、代理和通信机制等。一个良好的 API 或 SDK 是解决问题的关键。

从网格虚拟组织的角度出发，首先需要一个统一的身份认证方式，以一致的方法表示虚拟组织中的实体，比如使用符合 X.509v3 标准的证书。其次，可以与不同的本地管理域进行互操作，对组织中的资源可以进行统一的安全管理，比如实施统一的安全策略和访问控制。同时，一个网格计算任务通常由来自不同域的进程组成，在任务运行期间可以动态地变化，虚拟组织必须支持动态的安全组织通信。最后，网格还要支持不同的安全实现，这些技术可以是基于公钥体系的，也可以是基于对称密钥体系的，但是不应该依赖于某一种安全机制。

网格计算不仅需要解决普遍存在于 Internet 上的安全问题，还需要解决网格计算特有的安全问题，这为网格计算提出了一系列新的安全需求。网格安全技术的目的就是授予合法用户访问数据和执行操作的权限，防止用户非法操作或因操作失误造成数据泄密；防止数据被非法访问和修改或合法用户伪造数据进行欺骗行动，并且可以对用户进行审计和记费等。

具体来说，网格安全需求至少应包括如下部分。

- 认证需求：一站式认证、代理、协同认证、资源认证、基于用户的信任关系。
- 通信保护需求：灵活的信息保护策略、支持各种可靠的通信协议、支持独立数据单元的安全通信。
- 授权需求：资源所有者授权、限制代理。
- 灵活的安全策略：互操作性、名称映射、用户可选的安全策略、证书安全策略等。

网格系统及其应用需要提供标准的安全服务，包括认证、访问控制、数据完整性、机密性和抗抵赖性等。

10.4.3 网格的安全性需求及其安全架构

从用户的角度出发，网格安全面临的最基本的问题是如何在网格这样的复杂环境中提供给用户简单易用的安全功能。比如单点登录，一个用户只需要在提交网格计算任务前进行一次认证，以后在任务运行时，安全机制就可以在任务申请其他资源时进行自动认证，不需要用户再次参与认证过程。此外，用户证书和私有密钥必须得到安全而灵活的保护，使用户在任何地方都可以安全地使用自己的证书和密钥。最后，网格的安全机制应该对用户透明。

资源提供者则更关心如何控制和保护自己的资源。比如，虽然资源是提供给网格虚拟组织共享的，但是资源提供者有能力决定资源的本地访问控制策略，从而改变资源在虚拟组织中的访问控制。另外，本地安全系统和虚拟组织的协调和整合也是一个重要问题。取代或是修改这些安全系统是不现实的，只能通过映射和代理机制予以解决。

对于开发者而言，他们需要的是一个灵活和功能丰富的函数库，以方便地实现认证，灵活地实现消息保护、代理和通信机制等。一个良好的 API 或 SDK 是解决问题的关键。

从网格虚拟组织的角度出发，首先需要一个统一的身份认证方式，以一致的方法表示虚拟组织中的实体，比如使用符合 X.509v3 标准的证书。其次，可以与不同的本地管理域进行互操作，对组织中的资源可以进行统一的安全管理，比如实施统一的安全策略和访问控制。同时，一个网格计算任务通常由来自不同域的进程组成，在任务运行期间可以动态地变化，虚拟组织必须支持动态的安全组织通信。最后，网格还要支持不同的安全实现，这些技术可以是基于公钥体系的，也可以是基于对称密钥体系的，但是不应该依赖于某一种安全机制。

网格计算不仅需要解决普遍存在于 Internet 上的安全问题，还需要解决网格计算特有的安全问题，这为网格计算提出了一系列新的安全需求。网格安全技术的目的就是授予合法用户访问数据和执行操作的权限，防止用户非法操作或因操作失误造成数据泄密；防止数据被非法访问和修改或合法用户伪造数据进行欺骗行动，并且可以对用户进行审计和记费等。

具体来说，网格安全需求至少应包括如下部分。

- 认证需求：一站式认证、代理、协同认证、资源认证、基于用户的信任关系。
- 通信保护需求：灵活的信息保护策略、支持各种可靠的通信协议、支持独立数据单元的安全通信。
- 授权需求：资源所有者授权、限制代理。
- 灵活的安全策略：互操作性、名称映射、用户可选的安全策略、证书安全策略等。

网格系统及其应用需要提供标准的安全服务，包括认证、访问控制、数据完整性、机密性和抗抵赖性等。

根据网格的安全需求，图 10.12 给出了一个网格安全的总体构架。

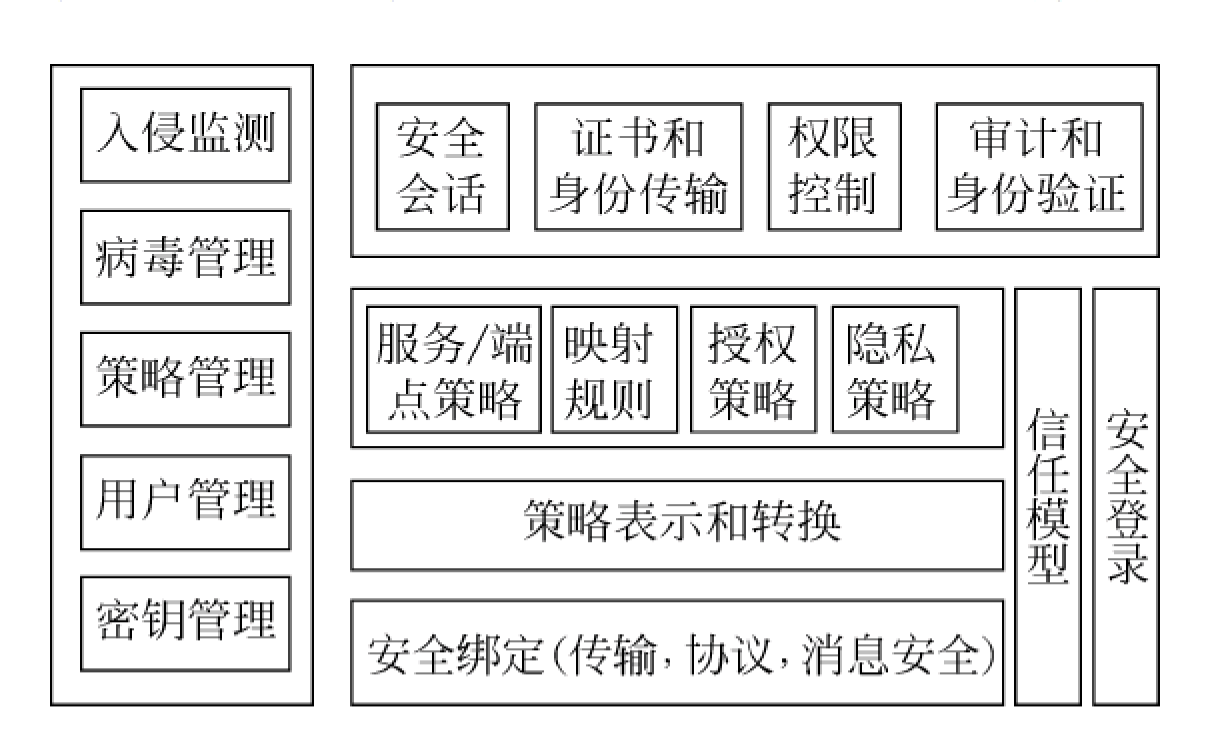


图 10.12 网格安全总体结构

10.5 网格安全基础设施

10.5.1 GSI 概述

GSI (grid security infrastructure)是 Globus 项目中针对 OGSA 的网格安全基础设施的实现，它使用了最常见的安全标准和设施。GSI 基于公钥加密体系（public key infrastructure，PKI)，采用 X.509 认证和安全套接字层通信协议，并对它们进行了一定的扩展，使得 GSI 可以支持单点登录。GSI 的实现符合通用安全服务编程接口(generic security service API，GSS-API)，GSS-API 是由 IETF 提出的用于安全系统的标准 API。GSI 能够提供的主要安全功能有安全认证、通信加密、私钥保护、委托授权和单点登录等。

1. 安全认证

安全认证是对服务请求者和服务提供者双方的身份进行验证的过程，GSI 中可以通过 SSL 实现。一个成功的安全认证，将校验一个请求连接的合法性，为其后双方的通信过程提供一个会话密钥。GSI 的安全认证基于用户的私钥创建一个代理，从而为用户提供认证方法。用户如果没有创建这个代理，就不能提交作业，也不能传输数据。

2. 通信加密

GSI 利用数字证书进行通信实体的相互认证，并通过 SSL/TLS 实现对数据的加密，以保证通信的安全。Globus Toolkit 中包含 OpenSSL，用于在网格客户机和服务器之间创建加密的通道。通信加密在安全认证之后，由通信双方产生一个会话密钥，并通过这个会话密钥加密通信过程中的消息。

3. 主机证书和私钥的存储与保护

GSI 将 Globus 用户的私钥保存在本地计算机的一个文件中。为了防止本地计算机的其他用户窃取私钥,该文件必须经过用户密码进行加密保护。GSI 还可以采用外部介质,如加密的智能卡来保存密钥,以更有效地保护用户私钥。

4. 委托授权

当用户与服务器认证成功后,将被委托授权。通过将用户 DN 号(唯一的证书主体名称)映射到本地用户账号,使网格用户的作业由这个本地用户根据自己的权限处理。

5. 单点登录

单点登录的目标是让用户只需输入一次密码,就可以完成多次认证,并且允许进程代表用户来进行资源的申请,这样用户的操作可以得到很大的简化。从用户的角度看,单点登录机制是用户在特定的逻辑安全域中,只需进行一次登录即可访问在此逻辑安全区域中不同应用系统中的被授权的资源,当超越了安全区域边缘时才需要再次登录。

GSI 以 X.509 证书实现认证,并通过对 X.509 证书进行扩展,产生代理证书来实现单点登录。

10.5.2 GSI 关键技术

1. 单点登录

GSI 中,单点登录是通过扩展 SSL 来实现的,这些扩展的功能包括代理证书(proxy credentials)和证书委托(credential delegation)。

代理证书由用户来签署,而不是由认证中心来签署。代理证书主要包含用户的身份标识、可以替换的私钥和证书的有效期等。为了简化频繁的访问操作,代理证书的存储通常不需要加密。代理证书即使被窃取,用户的私钥也不会泄露,而且证书的有效期限可以限制证书在被窃取后所造成的影响。用户输入一次密码,使用自己的数字证书产生代理证书后,在代理证书的有效期内,用户使用代理证书进行认证,就可在特定的逻辑安全区域中多次访问不同的数据资源,而不需要多次进行身份鉴别。

证书委托(certificate delegation)是指用户创建并委托代理证书给运行在远端资源上的进程授权,允许远程进程代表用户进行资源访问。证书委托提高了资源请求的效率,适合用于复杂的网格计算任务。

为了实现这两项功能,可以为用户创建一个用户代理,并且用户代理又可以在新的节点上创建新的代理,如此就可以在不同的节点之间形成一个安全信任链,如图 10.13 所示。

使用用户代理进行相互认证的扩展过程与标准的 SSL 认证过程的主要区别在于,被请

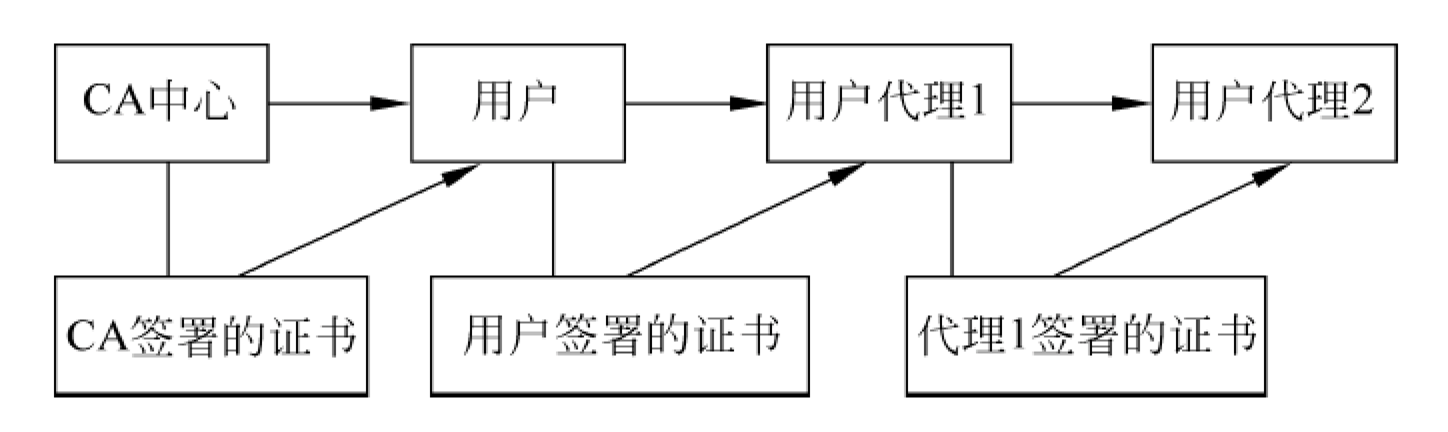


图 10.13 安全信任链

求方将会同时收到用户的数字证书(包含用户的公钥)和用户的代理数字证书。用户数字证书的公钥负责验证用户代理证书上的数字签名的合法性,这样实际上形成了一个认证中心—用户—代理的信任链。

用户代理使得用户可以做到只需输入一次密码就可以进行多次认证,从而实现单点登录。单点登录的认证过程如图 10.14 所示,从图中可以看出 GSI 在网格环境中的作用。图中各部分的执行步骤说明如下。

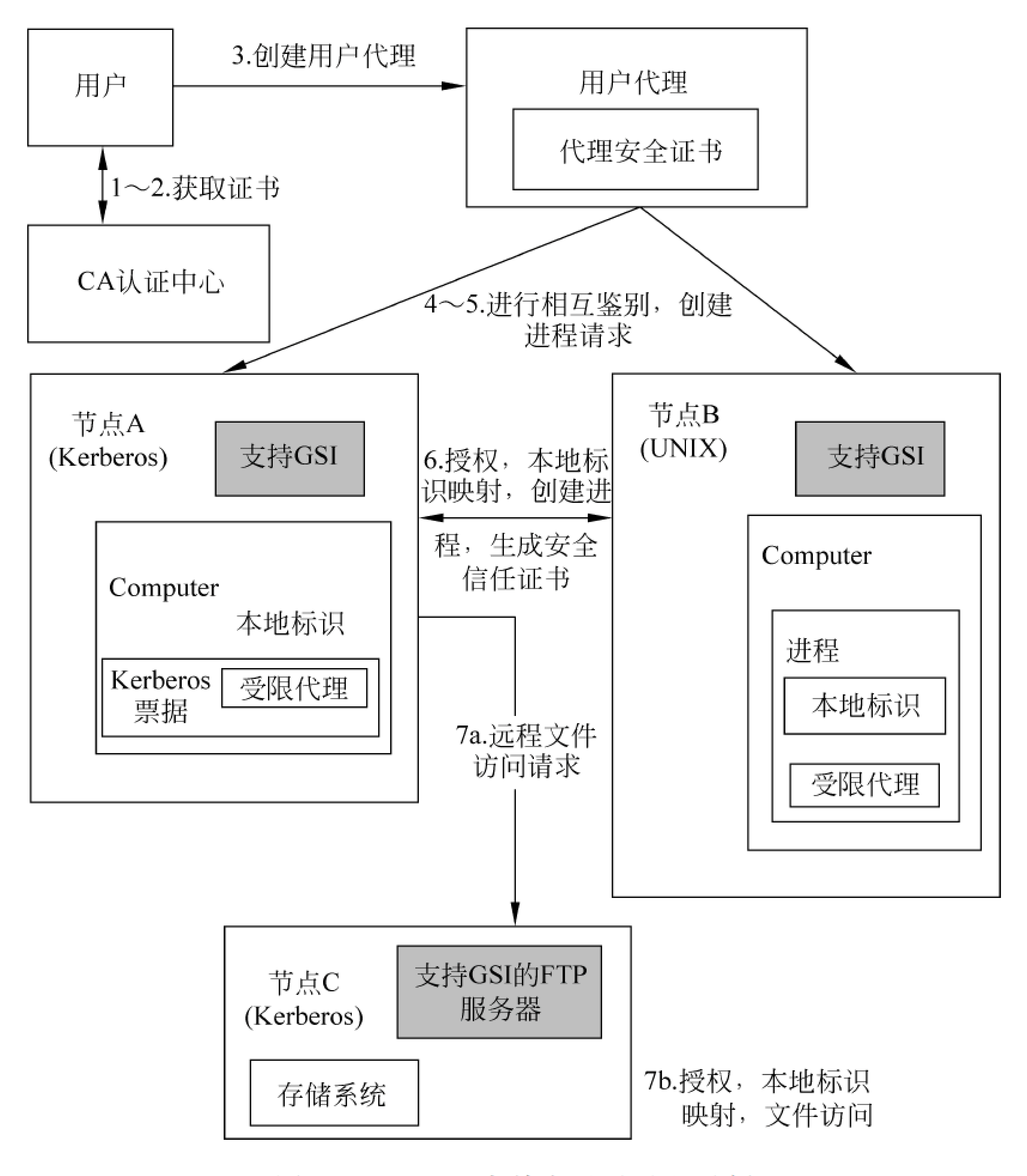


图 10.14 GSI 中单点登录过程示例

① 在进行 Globus 任务提交与执行之前，用户与服务节点和服务设施需要获得安全证书。Globus GSI 中，可以使用命令行命令 grid-cert-request 或对应的安全函数创建用于安全鉴别的公钥、私钥和未签发的安全证书，然后通过 E-mail 或其他安全途径把它们提交给 CA。

② CA 收到签发安全证书的请求后，对用户或服务节点进行审核，审核通过后，把签署过的安全证书返回给请求方。

③ 用户在提交任务前，可通过命令行命令 grid-proxy-init 或对应的安全函数创建一个临时的、局部的、有时间期限的用户代理。用户代理的安全证书由用户签署，并指定证书的有效期。

④ 用户代理与远端服务节点之间进行相互安全鉴别，即对双方的安全证书和身份进行鉴别。

⑤ 通过安全鉴别后，用户代理创建任务进程，并把任务提交给服务节点，服务节点再把任务提交给任务管理者进行具体处理。

⑥ 如果任务在执行过程中需要其他远程资源，也必须在任务进程与资源代理之间进行相互安全鉴别，通过安全鉴别后，任务进程才可以使用所需资源。

⑦ 如果任务在执行过程中需要访问远端数据或文件，也必须在任务进程与远端文件服务资源代理之间进行相互安全鉴别。通过安全鉴别后，还要进行授权、本地 ID 映射后，任务进程才可以访问远端数据或文件。

当任务执行完毕后，用户可以通过 grid-proxy-destroy 或对应的函数撤销用户代理。

2. 在线信任证存储

在线信任证存储 MyProxy 是为了方便用户从 Web 上登录网格而设立的。它为访问网格资源提供了一个公用的客户端代理。MyProxy 的结构与 DAC(自主访问控制)类似，都是将重要的信任证放置在一台中心服务器上，并对用户的登录进行认证。与 DAC 不同的是，DAC 直接对资源的访问者进行认证，而在 MyProxy 中，当 MyProxy 对用户完成鉴别后，MyProxy 将会给用户签发一个代理文档，然后用户再持此代理文档去访问远程资源。具体流程如图 10.15 所示。

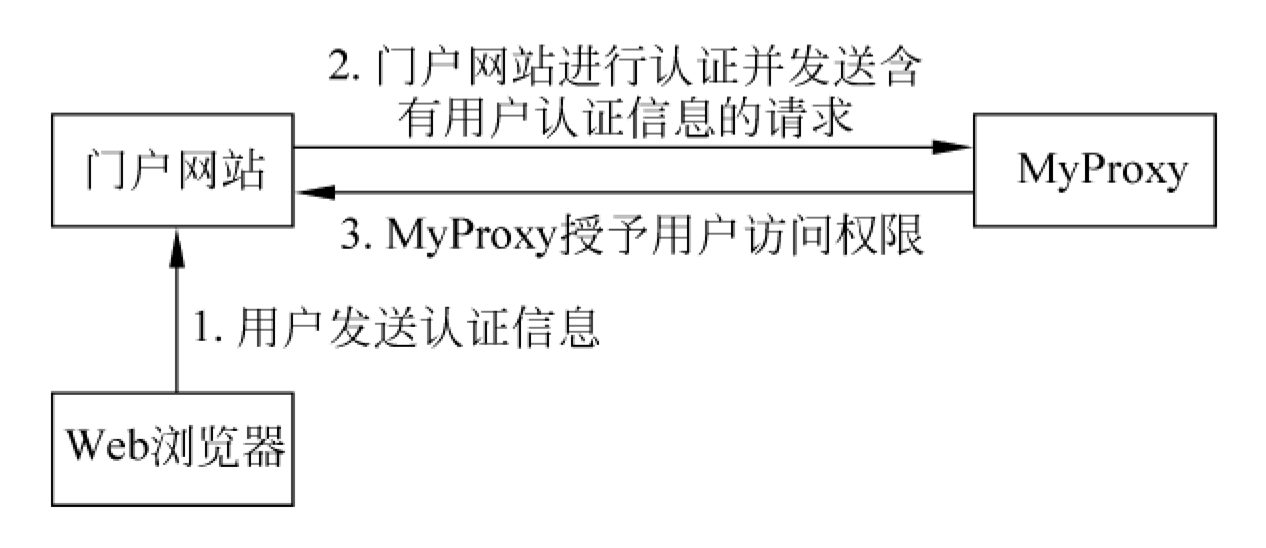


图 10.15 MyProxy 流程

MyProxy 的好处是显而易见的。它屏蔽了用户的位置,从而增强了网格部署的可扩展性。同时,MyProxy 被设计成高安全性服务器,可以提供对用户关键性信任证(信任信息)的保护。

MyProxy 机制也存在一些不足,例如,采用集中式结构,服务器面临单点失效的风险;只能根据用户密码签发代理证书,不能根据用户原有的证书签发新的证书;策略静态性较强、功能简单。

3. 虚拟组织与社区授权服务

如图 10.16 所示,虚拟组织是建立在多个物理组织之上的一个抽象的管理层。物理组织作为相对底层的管理者,负责制定资源的最终访问控制策略将一些资源共享给 VO,VO 负责高层的资源管理,制定共享资源在 VO 中的访问控制策略。

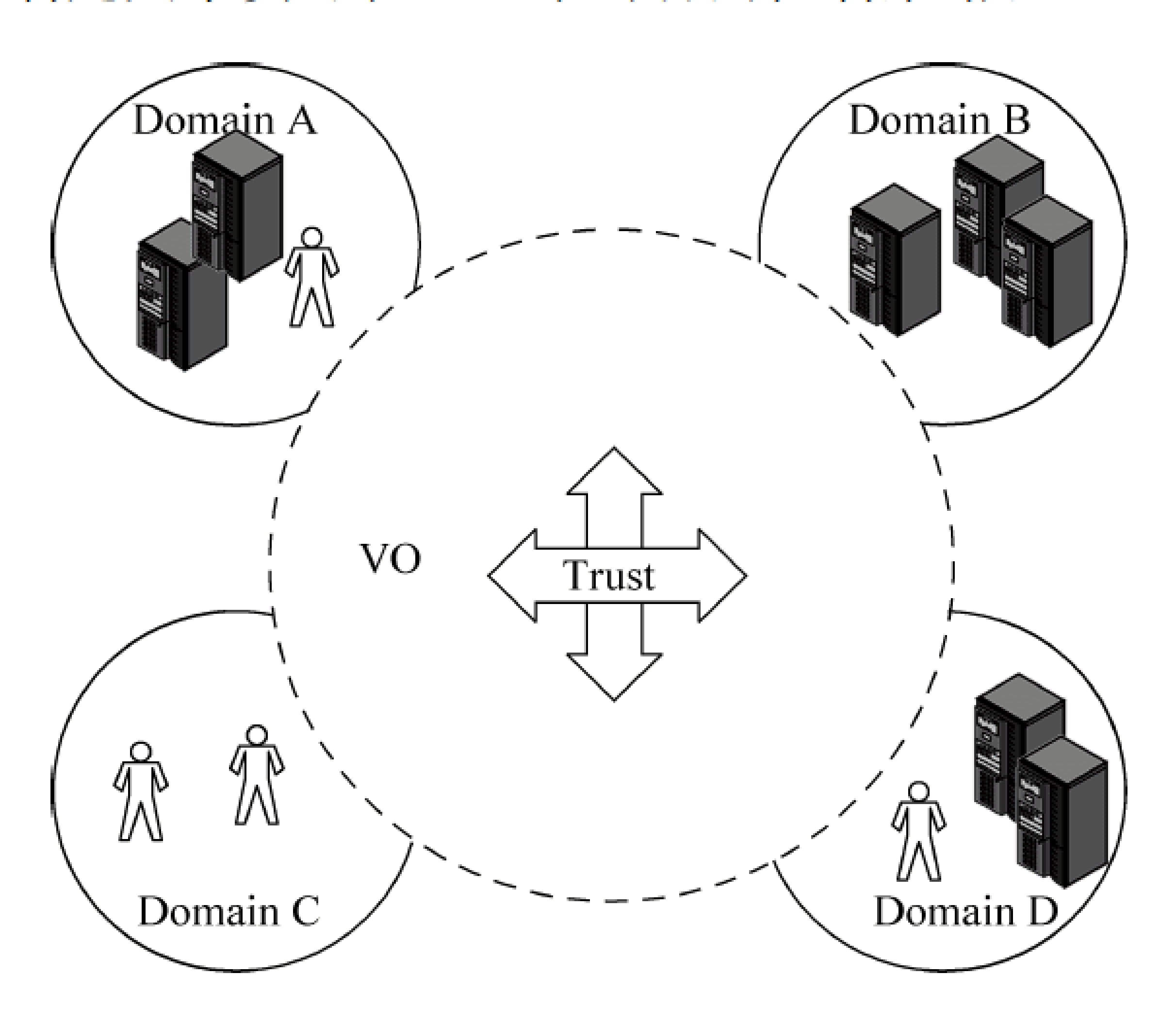


图 10.16 虚拟组织

由于 VO 的资源分布在多个不同的组织中,想要维持一个一致的策略就意味着每个组织都要执行一致的访问控制。由于每个站点都具有不同的策略表示和执行机制,并且这些机制可能在访问控制的粒度上是不同的。此外,VO 中的资源是动态的,VO 的策略也是动态的。因此,在 VO 中进行一致的访问控制不易实现。

社区授权服务(community authorization service,CAS)的提出是为了将资源授权的部分权力转交到网格中一些社区(community)上。这些团体控制着部分用户的认证和访问策略,使得原本单一的全局策略空间可以被划分为若干个以 CAS 作为桥梁的局部策略空间。

CAS 允许虚拟组织维护它自己的策略,并且可以使用这些策略与本地站点交互。站点则将本地策略(负责 VO 可以在本地做什么)和 VO 策略(负责 VO 成员在 VO 中能做什么)合并在一起,然后在本地执行这个合并的策略。VO 通过管理 CAS 服务器来维护合并后的

策略的 VO 部分，这部分策略包括如下方面。

① VO 对其资源的访问控制策略。

② CAS 服务器自身的访问控制策略，主要负责管理服务器和 VO 成员，如哪些用户可以添加成员到某个组。

③ VO 的成员列表。

合并策略的另一部分由资源提供者使用本地机制来维护。例如，一个站点可以通过映射的方法创建一个本地账号来代表一个 VO（一个站点可以属于多个 VO），然后将 VO 作为一个整体制定本地策略。

如图 10.17 所示，资源提供者将资源分配给 VO，并提供支持 CAS 的服务接口。用户访问这些资源时，首先需要与 VO 的 CAS 服务器建立连接，并请求一个 CAS 证书。CAS 服务器返回给用户一个经过 CAS 签名的证书，证书可以包含限制用户权限的策略。

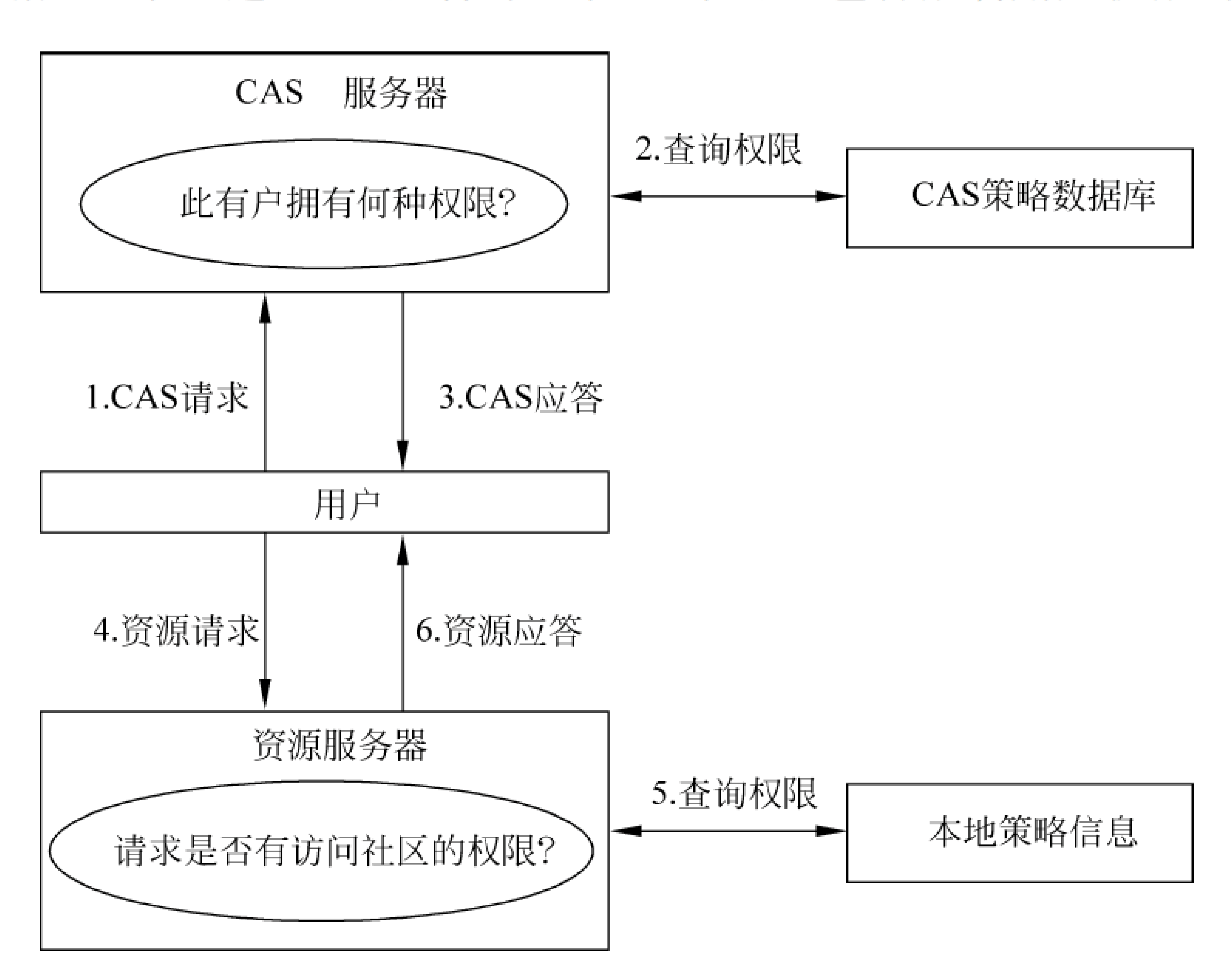


图 10.17 CAS 中的资源访问过程

用户可以使用这个证书和资源进行相互认证，认证完成后站点通过以下几个步骤来执行 VO 和本地策略。

① 检查 CAS 证书的有效性，如签名和有效期。

② 将 CAS 证书映射为本地 ID 或账号，然后执行站点对该账号的策略。

③ 执行 CAS 中 VO 对用户制定的策略。

④ 选择性地执行一些站点策略。

思考题

1. 实现单点登录的关键技术是什么？
2. 代理证书和普通证书有什么区别？
3. SAML 和 XACML 的区别和联系是什么？
4. 网格安全中的特殊需求是什么？GSI 平台中使用的关键技术是什么？

参 考 文 献

[1] Audin G. NEXT-Gen Firewalls: What to Expect. Business Communications Review Publication, June 2004

[2] Barker W. Introduction to the Analysis of the Data Encryption Standard(DES). Laguna Hills, CA: Aegean Park Press, 1991

[3] Bellare M, Rogeaway P. Collision Resistant Hashing: Towards Making UOWHT's Practical. Proceedings CRYPTO'97, 1997

[4] Bishop M. Computer Security: Art and Seience. Boston: Addison-Wesley, 2003

[5] Bishop M. Introduction to Computer Security. Boston: Addison-Wesley, 2005

[6] Vaudenay, Serge A. Classical Introduction to Cryptography: Applications for Communications Security, Springer, 2006

[7] Ray Hunt. Technological Infrastructure for PKI and Digital Certification. Computer Communications, 2005

[8] OpenSSL for Windows Developer's Guide. http://www. trizen. com, 2003

[9] John Linn. Trust Models and Management in Public-Key Infrastructures. RSA Laboratories, 2004

[10] Greg Shipley. Tactical Security 101. Network Computing, 2003(9)

[11] U. Blementhal, B. Wijnen. User-based Security Model (USM) for Version3 of Simple Networks Management Protocols. Journal of Information Security, 1999(11)

[12] CERT Coordination Center. Denial of Service Attacks. http://www. cert. org/tech_tips/denial_of_service. html

[13] Cheng P. A Security Architecture for the Internet Protocol. IBM SYSTEMS Journal, 1998

[14] Carrett P. Making, Breaking Codes: an Introduction to Cryptology. Upper Saddle river, NJ: Prentice Hall, 2001

[15] Pieprzyk J. Fundamentals of Computer Security. New York: Springer-Verlag, 2003

[16] Ludovic Me and Cedric Michel. Intrusion Detection: A Bibliography. Technical Report SSIR-2001-01, 2001

[17] Huang H. A Mandatory Access Control Model for Collaborative Environment. Journal of Harbin Institute of Technology. Vol 14 SUP, January 2007

[18] Hu L, Evans D. Secure Aggregation for Wireless Networks. Workshop on Security and Assurance in Ad Hoc Networks, January 2003

[19] Miller S K. Facing the Challenges of Wireless Security. IEEE Computer. 2001

[20] Michael E Whitman, Herbert J. Mattord. Principles of Information Security. Canada: GEX Publishing Serviees, 2003

[21] Nishio, Shuiehi. Standardization of Evaluation Criteria for IT security. Review, 2000(12)

[22] Schneier B. Cryptograph design and vulnerabilities. Computer, 1998, 31(9)

[23] Joshi J, Ghafoor A. Digital Government Security Infrastructure Design Challenges. Computer, 2001(34)

[24] David Johnston, Jesse Walker. Overview of IEEE 802. 16 Security. IEEE Security and Privacy, 2004

[25] Lee C H, Hwang M S, Yang W P. Enhanced Privacy and Authentication for the Global System for

Mobile Communications. Wireless Networks. 1999
[26] Stephan Northcutt, Judy Novak, Donald Mclanchian. Network Intrusion Detection Analyst's Handbook. Indiana: New Riders Publishing, 2000
[27] Derek Atkins. RFC3022: Internet Security Professional Reference. 1998
[28] Srisuresh P, Egevang K. Traditional IP Network Address Translator. 2001
[29] Sven Dietrich, Neil Long, Daivd Dittrich. Analyzing Distributed Denial of Service Attack Tools: The Shaft Case. In 14th Systems Administration Conference LISA, 2000
[30] P. Hoffman. RFC2406: Algorithms for Internet Key Exchange Version 1(IKEvl), May 2005
[31] P. Hoffman: RFC3664: The AES-XCBC-PEF-128 Algorithm for the Internet Key Exchange Protocol (IKE), January 2004
[32] Fanke R Glenn. RFC3602: The AES-CBC Cipher Algorithm and Its Use with IPSec, September 2003
[33] Carugi M, MeDysan D. RFC4031: Serviee Requirements for Layer3 Provider Provisioned Virtual Private Networks(PPVPNs), April 2005
[34] Santesson S, Housley R. RFC4325: Internet X. 509 Public Key Infrastructure Authority Information Access Certificate Revocation List (CRL) Extension, December 2005
[35] Schaad J. RFC4211: Internet X. 509 Public Key Infrastructure Certificate Request Message Format (CRMF), September 2005
[36] B. Ramsdell. RFC3851: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3. 1 Message Specification, July 2004
[37] T. Dierks, E. Rescorla. RFC4346: The Transport Layer Security (TLS) Protocol Version 1. 1, April 2006
[38] ATUL KAHA. 密码学与网络安全. 北京: 清华大学出版社, 2005
[39] GERT DE LAE. 网络安全基础. 张耀疆, 李磊译. 北京: 人民邮电出版社, 2006
[40] William Stallings. 密码编码学与网络安全——原理与实践(第四版)(英文影印版). 孟庆树译. 北京: 电子工业出版社, 2006
[41] William Stallings. 网络安全基础——应用与标准(第3版). 白国强译. 北京: 清华大学出版社, 2007
[42] Merike Kao. 网络安全设计. 吴中福译. 北京: 人民邮电出版社, 2005
[43] Ogletree Twzh. 防火墙原理与实施. 李之棠等译. 北京: 电子工业出版社, 2001
[44] Goncalvesm. 防火墙技术指南. 宋书名, 朱智强, 徐开勇译. 北京: 机械工业出版社, 2000
[45] Charlie Kaufman, Radia Perlman, Mike Speciner. 网络安全——公众世界中的秘密通信(第二版). 许剑卓, 左英男译. 北京: 电子工业出版社, 2004
[46] Douglas R. Stinson. 密码学原理与实践(第二版). 冯登国译. 北京: 电子工业版社, 2003. 9
[47] GREG HOLDEN. 防火墙与网络安全——入侵检测和VPN. 王斌, 孔璐等译. 北京: 清华大学出版社, 2004
[48] Man Young Rhee. 网络安全——加密原理、算法与协议. 金名, 张长富译. 北京: 清华大学出版社, 2007
[49] 寺田真敏, 萱岛信. TCP/IP网络安全篇, 王庆译. 北京: 科学出版社, 2003
[50] 赵泉. 网络安全与电子商务. 北京: 清华大学出版社, 2005
[51] 胡道元, 闵京华. 网络安全. 北京: 清华大学出版社, 2004
[52] 刘建伟, 张卫东, 刘培顺, 李晖. 网络安全实验教程. 北京: 清华大学出版社, 2007
[53] 冯登国. 计算机通信网络安全. 北京: 清华大学出版社, 2001

[54] 卿斯汉.密码学与计算机网络安全.北京：清华大学出版社,2001
[55] 张敏波.网络安全实战详解(企业专供版).北京：电子工业出版社,2008
[56] 龚俭,陆晟,王倩.计算机网络安全导论.南京：东南大学出版社,2000
[57] 麦肯兰勃.网络安全评估.北京：中国电力出版社,2004
[58] Mohan Atreya.数字签名.北京：清华大学出版社,2003
[59] 葛秀慧,田浩,金素梅.计算机网络安全管理(第二版).北京：清华大学出版社,2008
[60] 张仁斌,谭三,易勇,蒋毅.网络安全技术.北京：清华大学出版社,2004
[61] 刘建伟,王育民.网络安全——技术与实践.北京：清华大学出版社,2005
[62] 关振胜.公钥基础设施 PKI 与认证机构 CA.北京：电子工业出版社,2002.1
[63] 方勇,刘嘉勇.信息系统安全导论.北京：电子工业出版社,2003
[64] 关义章,戴宗坤.信息系统安全工程学.北京：电子工业出版社,2002
[65] 沈昌祥.信息安全工程导论.北京：电子工业出版社,2003
[66] 赵洪彪.信息安全策略.北京：清华大学出版社,2004
[67] 胡建伟.网络安全与保密.西安：西安电子科技大学出版社,2003
[68] 关振胜.公钥基础设施 PKI 与认证机构 CA.北京：电子工业出版社,2002
[69] 范红,冯登国.安全协议理论和方法.北京：科学出版社,2003
[70] 杨波.网络安全理论与应用.北京：电子工业出版社,2002
[71] 杨庚,沈剑刚,容淳铭.基于角色的访问控制理论研究.南京：南京邮电大学学报(自然科学版),2006.6
[72] 魏利明,陈相宁.PKI 技术分析.北京：网络安全技术与应用,2005.3
[73] 张仕斌,何达克,代群.PKI 安全认证体系的研究.北京：计算机应用研究,2005
[74] 冯登国,张阳,张玉清.信息安全风险评估综述.北京：通信学报,2004.25(7)
[75] 姚小兰.网络安全管理与防护.北京：北京理工大学出版社,2002.5
[76] 林闯,汪洋,李泉.网络安全的随机模型方法与评价技术.北京：计算机学报,2005(12)
[77] 徐超汉,柯宗贵.计算机网络安全实用技术(第二版).北京：电子工业出版社,2005.3
[78] 潘志翔,岑进锋.黑客攻防编程解析.北京：机械工业出版社,2003.1
[79] 叶月.网络安全实用技术.北京：清华大学出版社,2002.7
[80] 冯登国.计算机网络与通信安全(第二版).北京：清华大学出版社,2001.5
[81] 龚俭,陆最,王倩.计算机网络安全导论.南京：东南大学出版社,2002.6
[82] 肖军模,刘军,张炜,郝嘉林,周海刚.网络信息安全.北京：机械工业出版社,2003.9
[83] 梁煌编.计算机网络技术基础教程(第二版).北京：清华大学出版社,2004
[84] 胡建伟.网络安全与保密(第四版).西安：西安电子科技大学出版社,2003

读者意见反馈

亲爱的读者：

感谢您一直以来对清华版计算机教材的支持和爱护。为了今后为您提供更优秀的教材，请您抽出宝贵的时间来填写下面的意见反馈表，以便我们更好地对本教材做进一步改进。同时如果您在使用本教材的过程中遇到了什么问题，或者有什么好的建议，也请您来信告诉我们。

地址：北京市海淀区双清路学研大厦 A 座 602 室　计算机与信息分社营销室　收

邮编：100084　　电子邮件：jsjjc@tup.tsinghua.edu.cn

电话：010-62770175-4608/4409　　邮购电话：010-62786544

教材名称：计算机网络安全——协议、技术与应用

ISBN 978-7-302-18057-9

个人资料

姓名：________ 年龄：______ 所在院校/专业：________

文化程度：________ 通信地址：________

联系电话：________ 电子信箱：________

您使用本书是作为：□指定教材 □选用教材 □辅导教材 □自学教材

您对本书封面设计的满意度：

□很满意 □满意 □一般 □不满意　改进建议________

您对本书印刷质量的满意度：

□很满意 □满意 □一般 □不满意　改进建议________

您对本书的总体满意度：

从语言质量角度看　□很满意 □满意 □一般 □不满意

从科技含量角度看　□很满意 □满意 □一般 □不满意

本书最令您满意的是：

□指导明确 □内容充实 □讲解详尽 □实例丰富

您认为本书在哪些地方应进行修改？（可附页）

____

____

您希望本书在哪些方面进行改进？（可附页）

____

____

电子教案支持

敬爱的教师：

为了配合本课程的教学需要，本教材配有配套的电子教案（素材），有需求的教师可以与我们联系，我们将向使用本教材进行教学的教师免费赠送电子教案（素材），希望有助于教学活动的开展。相关信息请拨打电话 010-62776969 或发送电子邮件至 jsjjc@tup.tsinghua.edu.cn 咨询，也可以到清华大学出版社主页（http://www.tup.com.cn 或 http://www.tup.tsinghua.edu.cn）上查询。